

LilyPond

Le système de gravure musicale

Manuel de notation

L'équipe de développement de LilyPond

Ce document constitue le manuel de notation de GNU LilyPond 2.16.2. Sa lecture requiert une familiarité avec le contenu présenté dans le **Section “Manuel d’initiation”** dans *Manuel d’initiation*.

Pour connaître la place qu’occupe ce manuel dans la documentation, consultez la page **Section “Manuels”** dans *Informations générales*.

Si vous ne disposez pas de certains manuels, la documentation complète se trouve sur <http://www.lilypond.org/>.

Copyright © 1999–2012 par les auteurs.

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

La traduction de la notice de droits d’auteur ci-dessous vise à faciliter sa compréhension par le lecteur non anglophone, mais seule la notice en anglais a valeur légale.

Vous avez le droit de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU de documentation libre, version 1.1 ou tout autre version ultérieure publiée par la Free Software Foundation, “sans aucune section invariante”. Une copie de la licence est fournie à la section “Licence GNU de documentation libre”.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Pour LilyPond version 2.16.2

Table des matières

1	Notation musicale générale	1
1.1	Hauteurs	1
1.1.1	Écriture des hauteurs de note	1
	Hauteurs avec octave absolue	1
	Octaves relatives	2
	Altérations	5
	Nom des notes dans d'autres langues	7
1.1.2	Modification de plusieurs hauteurs	8
	Vérifications d'octave	9
	Transposition	10
	Inversion	13
	Rétrogradation	13
	Transformations modales	14
1.1.3	Gravure des hauteurs	16
	Clefs	16
	Armure	19
	Marques d'octaviation	21
	Instruments transpositeurs	22
	Altérations accidentelles automatiques	23
	Ambitus	30
1.1.4	Têtes de note	33
	Têtes de note spécifiques	33
	Têtes de note avec nom de note	34
	Têtes de note à forme variable	36
	Improvisation	39
1.2	Rythme	39
1.2.1	Écriture du rythme	40
	Durées	40
	Nolets	42
	Changement d'échelle des durées	46
	Liaisons de prolongation	47
1.2.2	Écriture des silences	51
	Silences	51
	Silences invisibles	53
	Silences valant une mesure	54
1.2.3	Gravure du rythme	58
	Métrique	58
	Indication métronomique	63
	Levées	66
	Musique sans métrique	67
	Notation polymétrique	69
	Découpage automatique des notes	72
	Gravure de lignes rythmiques	73
1.2.4	Barres de ligature	75
	Barres de ligature automatiques	75
	Définition des règles de ligature automatique	78
	Barres de ligature manuelles	86
	Liens de croches en soufflet	89

1.2.5	Mesures	90
	Barres de mesure	90
	Numéros de mesure	94
	Vérification des limites et numéros de mesure	99
	Indications de repère	100
1.2.6	Fonctionnalités rythmiques particulières	102
	Notes d'ornement	102
	Alignement et cadences	107
	Gestion du temps	108
1.3	Signes d'interprétation	109
1.3.1	Signes d'interprétation attachés à des notes	109
	Articulations et ornements	109
	Nuances	112
	Personnalisation des indications de nuance	117
1.3.2	Signes d'interprétation sous forme de courbe	119
	Liaisons d'articulation	119
	Liaisons de phrasé	122
	Signes de respiration	123
	Chutes et sauts	124
1.3.3	Signes d'interprétation sous forme de ligne	125
	Glissando	125
	Arpèges	129
	Trilles	132
1.4	Répétitions et reprises	134
1.4.1	Répétition d'un long passage	134
	Répétitions courantes	135
	Indications de reprise manuelles	140
	Répétitions explicites	143
1.4.2	Autres types de répétition	145
	Répétitions de mesure	145
	Répétitions en trémolo	147
1.5	Notes simultanées	149
1.5.1	Monophonie	149
	Notes en accords	149
	Répétition d'accords	151
	Expressions simultanées	153
	Clusters	154
1.5.2	Plusieurs voix	154
	Polyphonie sur une portée	155
	Styles de voix	158
	Résolution des collisions	158
	Regroupement automatique de parties	163
	Saisie de musique en parallèle	168
1.6	Notation sur la portée	170
1.6.1	Gravure des portées	171
	Initialisation de nouvelles portées	171
	Regroupement de portées	172
	Imbrication de regroupements de portées	176
	Séparation des systèmes	177
1.6.2	Modification de portées individuelles	178
	Symbole de la portée	178
	Portées d'ossia	181
	Masquage de portées	185
1.6.3	Écriture de parties séparées	188

Noms d'instrument	188
Citation d'autres voix	192
Mise en forme d'une citation	195
1.7 Annotations éditoriales	200
1.7.1 Dans la portée	200
Indication de la taille de fonte musicale	200
Doigtés	201
Dictée à trous	204
Coloration d'objets	205
Parenthèses	206
Hampes	207
1.7.2 Hors de la portée	207
Info-bulle	207
Quadrillage temporel	208
Crochets d'analyse	211
1.8 Texte	212
1.8.1 Ajout de texte	212
Commentaires textuels	213
Indication textuelle avec extension	214
Indications textuelles	216
Texte indépendant	219
1.8.2 Mise en forme du texte	221
Introduction au formatage de texte	221
Sélection de la fonte et de la taille	223
Alignement du texte	225
Éléments graphiques dans du texte formaté	228
Notation musicale dans du texte formaté	231
Texte avec sauts de page	233
1.8.3 Fontes	234
Tout savoir sur les fontes	234
Attribution d'une fonte en particulier	236
Choix des fontes par défaut	236
2 Notation spécialisée	238
2.1 Musique vocale	238
2.1.1 Vue d'ensemble de la musique vocale	238
Références en matière de musique vocale	238
Saisie des paroles	239
Alignement des paroles sur la mélodie	240
Durée automatique des syllabes	242
Durée explicite des syllabes	245
Plusieurs syllabes sur une note	246
Plusieurs notes pour une même syllabe	247
Traits d'union et de prolongation	250
2.1.2 Situations particulières en matière de paroles	250
Travail avec des paroles et variables	250
Positionnement vertical des paroles	252
Positionnement horizontal des syllabes	256
Paroles et reprises	258
Paroles alternatives	266
2.1.3 Couplets	267
Numérotation des couplets	267
Indication de nuance dans les couplets	268
Indication du personnage et couplets	268

Rythme différent selon le couplet	269
Paroles en fin de partition	272
Paroles sur plusieurs colonnes en fin de partition	273
2.1.4 Chansons	274
Références en matière de chanson	274
Feuille de chant	275
2.1.5 Chorale	275
Références en matière de chorale	275
Mise en forme d'une partition chorale	276
Division de voix	278
2.1.6 Opéras et musiques de scène	278
Références en matière d'opéra et musique de scène	279
Indication du rôle	279
Citation-repère	281
Musique parlée	285
Dialogue et musique	285
2.1.7 Chants liturgiques	287
Références en matière de chant liturgique	287
Cantiques et hymnes	287
Psalmodie	294
Mesure incomplète et musique liturgique	297
2.1.8 Musique vocale ancienne	299
2.2 Instruments utilisant des portées multiples	300
2.2.1 Vue d'ensemble des claviers	301
Généralités sur les instruments à clavier	301
Changement de portée manuel	302
Changement de portée automatique	303
Lignes de changement de portée	305
Hampes et changements de portée	305
2.2.2 Piano	307
Pédales de piano	307
2.2.3 Accordéon	308
Symboles de jeux	308
2.2.4 Harpe	312
Généralités sur la harpe	312
Pédales de harpe	312
2.3 Cordes non frettées	313
2.3.1 Vue d'ensemble de la notation pour cordes non frettées	314
Références en matière de cordes non frettées	314
Indications d'archet	314
Harmoniques	315
Snap (Bartók) pizzicato	316
2.4 Instruments à cordes frettées	316
2.4.1 Vue d'ensemble des cordes frettées	317
Références en matière de cordes frettées	317
Indications du numéro de corde	317
Tablatures par défaut	319
Tablatures personnalisées	331
Tablatures sous forme d'étiquette	334
Tablatures prédéfinies	344
Tablatures automatiques	354
Doigtés pour la main droite	357
2.4.2 Guitare	358
Indication de la position et du barré	358

Indication des harmoniques et notes étouffées	359
Indication de <i>power chord</i>	360
2.4.3 Banjo	362
Tablatures pour banjo	362
2.5 Percussions	362
2.5.1 Vue d'ensemble des percussions	362
Références en matière de notation pour percussions	363
Notation de base pour percussions	363
Roulements de tambour	364
Hauteurs en percussions	365
Portées de percussion	365
Personnalisation de portées de percussion	367
Notes fantômes	371
2.6 Instruments à vent	371
2.6.1 Vue d'ensemble des instruments à vent	372
Références en matière d'instruments à vent	372
Doigtés pour vents	373
2.6.2 Cornemuse	375
Définitions pour la cornemuse	375
Exemple pour la cornemuse	375
2.6.3 Bois	377
2.6.3.1 Diagrammes pour bois	377
2.7 Notation des accords	385
2.7.1 Mode accords	385
Généralités sur le mode accords	385
Accords courants	386
Extension et altération d'accords	388
2.7.2 Gravure des accords	390
Impression des noms d'accord	391
Personnalisation des noms d'accord	393
2.7.3 Basse chiffrée	399
Introduction à la basse chiffrée	399
Saisie de la basse chiffrée	400
Gravure de la basse chiffrée	403
2.8 Musique contemporaine	406
2.8.1 Hauteur et harmonie en musique contemporaine	406
Généralités en matière de hauteur et d'harmonie	406
Notation microtonale	406
Armures contemporaines et harmonie	406
2.8.2 Approches du rythme en musique contemporaine	406
Généralités sur le rythme en musique contemporaine	406
Nolets et musique contemporaine	407
Métriques contemporaines	407
Notation polymétrique étendue	407
Ligatures et musique contemporaine	407
Barres de mesure et musique contemporaine	407
2.8.3 Notation graphique	407
2.8.4 Techniques de partition contemporaine	407
2.8.5 Nouvelles techniques instrumentales	407
2.8.6 Informations complémentaires et exemples pertinents	407
Ouvrages et articles sur la notation en musique contemporaine	407
Partitions et exemples	407
2.9 Notations anciennes	407
Formes de notation ancienne prises en charge	409

2.9.1	Considérations communes aux musiques anciennes	409
	Contextes prédéfinis	409
	Ligatures	410
	Guidons	411
	Prise en charge de la basse figurée	411
2.9.2	Typographie de musique ancienne	411
	Contextes de musique mensurale	411
	Clefs anciennes	412
	Métriques anciennes	413
	Têtes de note anciennes	414
	Crochets anciens	415
	Silences anciens	416
	Altérations et armures anciennes	417
	Altérations suggérées (<i>musica ficta</i>)	417
	Ligatures mensurales	418
2.9.3	Typographie du chant grégorien	419
	Contextes du chant grégorien	419
	Clefs grégoriennes	420
	Altérations et armures grégoriennes	421
	Divisions	421
	Articulations grégoriennes	422
	Points d'augmentation (<i>morae</i>)	423
	Neumes et ligatures grégoriennes	423
2.9.4	Typographie de notation kiévienne	430
	Contextes de notation kiévienne	430
	Clefs kiéviennes	431
	Têtes de note kiéviennes	431
	Altérations kiéviennes	432
	Barre de mesure kiévienne	432
2.9.5	Réédition de musique ancienne	432
	Des incipits	433
	Mise en forme de la musique mensurale	433
	Transcription de chant grégorien	433
	Éditions ancienne et moderne à partir d'une même source	437
	Notation éditoriale	437
2.10	Musiques du monde	437
2.10.1	Noms des notes et altérations non-occidentaux	437
	Extension des systèmes de notation et d'accordage	437
2.10.2	Musique arabe	438
	Références pour la musique arabe	438
	Noms des notes en arabe	439
	Armures arabes	439
	Métriques arabes	441
	Exemple de musique arabe	442
	Lectures complémentaires pour la musique arabe	442
2.10.3	Musique classique turque	443
	Références pour la musique classique turque	443
	Noms de note en turc	443

3	Généralités en matière d'entrée et sortie	445
3.1	Agencement du code	445
3.1.1	Structure d'une partition	445
3.1.2	Plusieurs partitions dans un même ouvrage	446
3.1.3	Plusieurs éditions pour une même source	447
3.1.4	Nom des fichiers de sortie	448
3.1.5	Structure de fichier	449
3.2	Titres et entêtes	451
3.2.1	Création de titres et entête ou pied de page	451
	Généralités en matière de titrages	451
	Mise en forme par défaut des titrages subalternes	454
	Mise en forme par défaut des entête et pied de page	457
3.2.2	Titrages personnalisés	458
	Mise en forme personnalisée des champs de titrage	458
	Mise en forme personnalisée des titrages	459
	Mise en forme personnalisée des entête et pied de page	462
3.2.3	Notes de bas de page	463
	Notes de bas de page dans une expression musicale	464
	Notes de bas de page dans du texte indépendant	467
3.2.4	Référencement des numéros de page	470
3.2.5	Table des matières	471
3.3	Travail sur des fichiers texte	473
3.3.1	Insertion de fichiers LilyPond	473
3.3.2	Différentes éditions à partir d'une même source	475
	Utilisation de variables	475
	Utilisation de balises	477
	Globalisation des réglages	480
3.3.3	Caractères spéciaux	480
	Codage du texte	480
	Unicode	481
	Équivalents ASCII	482
3.4	Contrôle des sorties	483
3.4.1	Extraction de fragments musicaux	483
3.4.2	Ignorer des passages de la partition	483
3.4.3	Formats de sortie alternatifs	484
3.4.4	Changement des fontes musicales	484
3.5	Sortie MIDI	485
3.5.1	Création de fichiers MIDI	485
	Noms d'instrument	486
3.5.2	Le bloc MIDI	487
3.5.3	Contenu de la sortie MIDI	488
	Éléments pris en compte dans le MIDI	488
	Éléments non pris en compte dans le MIDI	488
3.5.4	Répétitions et MIDI	489
3.5.5	Gestion des nuances en MIDI	489
	Indications de nuance	489
	Amplitude du volume en MIDI	490
	Égalisation de plusieurs instruments (i)	491
	Égalisation de plusieurs instruments (ii)	492
3.5.6	MIDI et percussions	493
3.5.7	Le script Articulate	494
3.6	Extraction d'informations musicales	494
3.6.1	Affichage de notation au format LilyPond	494
3.6.2	Affichage de la musique sous forme d'expression Scheme	495

3.6.3	Enregistrement d'événements musicaux dans un fichier	495
4	Gestion de l'espace.....	496
4.1	Mise en forme de la page	496
4.1.1	Le bloc <code>\paper</code>	496
4.1.2	Format du papier et adaptation automatique.....	497
	Format du papier	497
	Adaptation automatique au format	498
4.1.3	Variables d'espacement vertical fixe	499
4.1.4	Variables d'espacement vertical fluctuant	500
	Structure des variables d'espacement vertical fluctuant	500
	Liste des variables d'espacement fluctuant	501
4.1.5	Variables d'espacement horizontal	502
	Variables de marge et de largeur	502
	Variables spécifiques pour l'impression recto-verso	503
	Variables d'indentation et de décalage	503
4.1.6	Autres variables du bloc <code>\paper</code>	504
	Variables de gestion des sauts de ligne	504
	Variables de gestion des sauts de page	504
	Variables de gestion des numéros de page	505
	Variables supplémentaires	506
4.2	Mise en forme de la partition	507
4.2.1	Le bloc <code>\layout</code>	507
4.2.2	Définition de la taille de portée	509
4.3	Sauts	510
4.3.1	Sauts de ligne	510
4.3.2	Sauts de page	512
4.3.3	Optimisation des sauts de page	513
4.3.4	Optimisation des tournes	513
4.3.5	Minimisation des sauts de page	514
4.3.6	Présentation en rouleau	514
4.3.7	Sauts explicites	515
4.3.8	Recours à une voix supplémentaire pour gérer les sauts	516
4.4	Espacement vertical	518
4.4.1	Espacement vertical au sein d'un système	518
	Propriétés d'espacement au sein d'un système	519
	Espacement de portées isolées	522
	Espacement de portées regroupées	523
	Espacement des lignes rattachées à des portées	524
4.4.2	Positionnement explicite des portées et systèmes	525
4.4.3	Résolution des collisions verticales	532
4.5	Espacement horizontal	533
4.5.1	Généralités sur l'espacement horizontal	533
4.5.2	Changement d'espacement en cours de partition	535
4.5.3	Modification de l'espacement horizontal	535
4.5.4	Longueur de ligne	537
4.5.5	Notation proportionnelle	538
4.6	Réduction du nombre de pages de la partition	545
4.6.1	Mise en évidence de l'espacement	546
4.6.2	Modification de l'espacement	547

5	Modification des réglages prédéfinis	549
5.1	Contextes d'interprétation	549
5.1.1	Tout savoir sur les contextes	549
	Score – le père de tous les contextes	549
	Contextes de haut niveau – les systèmes	549
	Contextes de niveau intermédiaire – les portées	550
	Contextes de bas niveau – les voix	550
5.1.2	Création d'un contexte	551
5.1.3	Conservation d'un contexte	552
5.1.4	Modification des greffons de contexte	555
5.1.5	Modification des réglages par défaut d'un contexte	557
	Modification de tous les contextes d'un même type	557
	Modification d'un contexte particulier	560
	Ordre de préséance	561
5.1.6	Définition de nouveaux contextes	562
5.1.7	Ordonnancement des contextes	564
5.2	En quoi consiste la référence des propriétés internes	565
5.2.1	Navigation dans les références du programme	565
5.2.2	Interfaces de rendu	566
5.2.3	Détermination de la propriété d'un objet graphique (grob)	567
5.2.4	Conventions de nommage	568
5.3	Modification de propriétés	569
5.3.1	Vue d'ensemble de la modification des propriétés	569
5.3.2	La commande de fixation <code>\set</code>	569
5.3.3	La commande de dérogation <code>\override</code>	571
5.3.4	La commande d'affinage <code>\tweak</code>	573
5.3.5	<code>\set</code> ou <code>\override</code>	575
5.3.6	Modification de listes associatives	575
5.4	Propriétés et contextes utiles	577
5.4.1	Modes de saisie	577
5.4.2	Direction et positionnement	579
5.4.3	Distances et unités de mesure	580
5.4.4	Propriétés des symboles de la portée	581
5.4.5	Extenseurs et prolongateurs	581
	Utilisation de <code>spanner-interface</code>	581
	Utilisation de <code>line-spanner-interface</code>	584
5.4.6	Visibilité des objets	586
	Suppression des stencils	586
	Transparence des objets	587
	Blanchiment des objets	587
	Utilisation de <code>break-visibility</code>	587
	Considérations spécifiques	589
5.4.7	Styles de ligne	591
5.4.8	Rotation des objets	592
	Rotation des objets de mise en forme	592
	Rotation des étiquettes	592
5.5	Retouches avancées	593
5.5.1	Alignement des objets	593
	Détermination directe de <code>X-offset</code> et <code>Y-offset</code>	594
	Utilisation de <code>side-position-interface</code>	594
	Utilisation de <code>self-alignment-interface</code>	594
	Utilisation de <code>break-aligned-interface</code>	596
5.5.2	Regroupement vertical d'objets graphiques	598
5.5.3	Modification des stencils	598

5.5.4	Modification de l'allure des éléments	599
	Modification des liaisons	599
5.5.5	Conteneurs requalifiants	600
5.6	Utilisation de fonctions musicales	602
5.6.1	Syntaxe d'une fonction de substitution	602
5.6.2	Exemples de fonction de substitution	603
Annexe A	Tables du manuel de notation	605
A.1	Table des noms d'accord	605
A.2	Modificateurs d'accord usuels	606
A.3	Accordages prédéfinis	609
A.4	Diagrammes d'accord prédéfinis	610
A.5	Formats de papier prédéfinis	616
A.6	Instruments MIDI	618
A.7	Liste des couleurs	619
A.8	La fonte Feta	620
	Glyphes de clef	620
	Glyphes de métrique	621
	Glyphes de chiffre	621
	Glyphes d'altération	621
	Glyphes de tête de note par défaut	622
	Glyphes de tête de note spéciale	623
	Glyphes de tête de note à forme variable	623
	Glyphes de silence	627
	Glyphes de crochet de croche	628
	Glyphes de point	628
	Glyphes de nuance	629
	Glyphes de script	629
	Glyphes de flèche	631
	Glyphes d'extrémité d'accolade	632
	Glyphes de pédale	632
	Glyphes d'accordéon	632
	Glyphes de liaison	632
	Glyphes de style vaticana	633
	Glyphes de style medicaea	634
	Glyphes de style Hufnagel	634
	Glyphes de style mensural	635
	Glyphes de style néomensural	638
	Glyphes de style Petrucci	639
	Glyphes de style Solesmes	640
	Glyphes de style kiévien	640
A.9	Styles de tête de note	641
A.10	Commandes pour markup	642
A.10.1	Font	642
A.10.2	Align	650
A.10.3	Graphic	665
A.10.4	Music	671
A.10.5	Instrument Specific Markup	675
A.10.6	Other	678
A.11	Commandes pour liste de markups	685
A.12	Liste des caractères spéciaux	686
A.13	Liste des signes d'articulation	687
	Scripts d'articulation	687
	Scripts d'ornement	687

Scripts de point d'orgue et point d'arrêt	688
Scripts spécifiques à certains instruments	688
Scripts de reprise et de répétition	688
Scripts pour musique ancienne	688
A.14 Notes utilisées en percussion	688
A.15 Glossaire technique	690
alist (liste associative)	690
callback (rappel)	690
closure (clôture)	690
glyphe	690
grob (objet graphique)	691
inaltérable	691
interface	691
lexer (analyseur lexical)	691
altérable	692
output-def (définition de sortie)	692
parser (analyseur syntaxique)	692
variable de l'analyseur grammatical	692
prob (objet de propriété)	693
clôture simple	693
smob (objet Scheme)	693
stencil	693
A.16 Liste des propriétés de contexte	693
A.17 Propriétés de mise en forme	704
A.18 Fonctions musicales prédéfinies	722
A.19 Identificateurs de modification de contexte	729
A.20 Types de prédicats prédéfinis	729
R5RS primary predicates	730
R5RS secondary predicates	730
Guile predicates	730
LilyPond scheme predicates	730
LilyPond exported predicates	731
A.21 Fonctions Scheme	732
Annexe B Aide-mémoire	755
Annexe C Grammaire de LilyPond	758
Annexe D GNU Free Documentation License	784
Annexe E Index des commandes LilyPond	791
Annexe F Index de LilyPond	801

1 Notation musicale générale

Ce chapitre explique comment créer la notation musicale standard.

1.1 Hauteurs

The image displays two systems of musical notation for piano. The first system consists of two staves (treble and bass clef) in C major, 4/4 time. It features a melody in the right hand and a harmonic accompaniment in the left hand. Dynamics include *p* (piano), *cresc.* (crescendo), and *sf* (sforzando). The phrase *dolce e molto legato* is written above the first staff. The second system, starting at measure 38, continues the piece with similar notation and dynamics. Below the staves, there are symbols for repeat signs and asterisks indicating specific musical instructions or errors.

Cette section détaille la façon d'indiquer la hauteur des notes, sous trois aspects : la saisie des hauteurs, la modification des hauteurs et les options de gravure.

1.1.1 Écriture des hauteurs de note

Cette section explique la manière d'indiquer les hauteurs de note. Deux modes permettent d'indiquer l'octave des notes : le mode absolu, et le mode relatif. Ce dernier est le plus pratique lors de la saisie d'un fichier source au clavier de l'ordinateur.

Hauteurs avec octave absolue

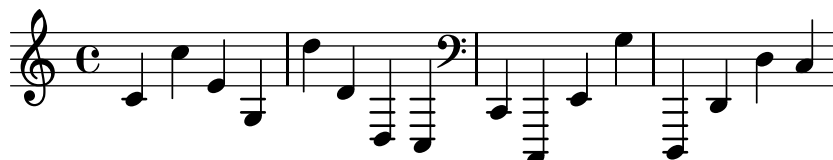
La hauteur s'écrit – à moins de préciser une autre langue – avec la notation batave, en utilisant les lettres de a à g. Les notes c (do) et b (si) sont écrites une octave sous le do central.

```
{
  \clef bass
  c4 d e f
  g4 a b c
  d4 e f g
}
```



L'octave peut être précisée sous forme d'une série d'apostrophes ' ou d'une série de virgules ,. Chaque ' hausse la note d'une octave ; chaque , baisse la note d'une octave.

```
{
  \clef treble
  c'4 c' ' e' g
  d''4 d' d c
  \clef bass
  c,4 c,, e, g
  d,,4 d, d c
}
```



Voir aussi

Glossaire musicologique : [Section “Noms des notes” dans *Glossaire*.](#)

Morceaux choisis : [Section “Hauteurs” dans *Morceaux choisis*.](#)

Octaves relatives

On spécifie les octaves en ajoutant ' ou , aux noms de hauteur. En recopiant de la musique, on a vite fait de mettre une note à la mauvaise octave ; ce genre d’erreur est difficile à retrouver. Le mode d’écriture en octave relative prévient ces erreurs dans la mesure où elles deviennent beaucoup plus évidentes : une seule erreur décale le reste de la pièce à une mauvaise octave.

`\relative hauteur_de_référence expression_musicale`

En mode relatif, chaque note est considérée comme étant le plus proche possible de celle qui la précède. L’octave des notes mentionnées dans *expression_musicale* va être calculée de la manière suivante :

- Si aucun signe de changement d’octave n’est utilisé, l’intervalle de base entre la note actuelle et la précédente sera toujours au plus d’une quarte. Cet intervalle est déterminé sans tenir compte des altérations.
- Un signe de changement d’octave ' ou , peut être ajouté pour hausser ou baisser la note d’une octave par rapport à la hauteur calculée sans spécification.
- Ces signes de changement d’octave peuvent être multipliés. Par exemple, '' ou ,, ajouteront une octave supplémentaire.
- Lorsque l’on entre en mode `\relative`, une hauteur absolue de départ peut être spécifiée, et agira dès lors comme si elle précédait la première note de *expression_musicale*. Nous vous recommandons de faire référence à un do (c).

Voici le mode `\relative` en action.

```
\relative c {
  \clef bass
  c d e f
  g a b c
  d e f g
}
```



On utilise les signes de changement d'octave pour les intervalles dépassant la quarte.

```
\relative c'' {
  c g c f,
  c' a, e'' c
}
```



Bien que ne comportant aucun signe de changement d'octave, une séquence de notes peut tout à fait couvrir un intervalle important.

```
\relative c {
  c f b e
  a d g c
}
```



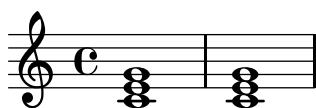
Lorsque plusieurs blocs `\relative` sont imbriqués, c'est la dernière clause `\relative` qui est prise en compte.

```
\relative c' {
  c d e f
  \relative c'' {
    c d e f
  }
}
```



`\relative` est sans effet sur un bloc `\chordmode`.

```
\new Staff {
  \relative c''' {
    \chordmode { c1 }
  }
  \chordmode { c1 }
}
```



`\relative` n'est pas permis au sein d'un bloc `\chordmode`.

Pour utiliser le mode d'octave relative dans de la musique transposée, une clause `\relative` additionnelle doit être placée au sein du bloc `\transpose`.

```

\relative c' {
  d e
  \transpose f g {
    d e
    \relative c' {
      d e
    }
  }
}

```



Si l'expression précédente est un accord, c'est la première note de l'accord qui détermine l'emplacement de la première note du prochain accord. À l'intérieur de l'accord, les notes sont placées relativement à celle qui précède. Examinez avec attention l'exemple suivant, et tout particulièrement le positionnement des do.

```

\relative c' {
  c
  <c e g>
  <c' e g'>
  <c, e, g' '>
}

```



Comme nous l'avons vu, l'octaviation est déterminée sans tenir compte des altérations. Ainsi un mi double-dièse qui suit un si naturel sera placé au-dessus de celui-ci, alors qu'un fa double-bémol se retrouvera en dessous. En d'autres termes, une quarte doublement augmentée demeure considérée comme un intervalle plus petit qu'une quinte diminuée, bien que la quarte doublement augmentée soit de sept demi-tons et la quinte diminuée de seulement six demi-tons.

```

\relative c'' {
  c2 fis
  c2 ges
  b2 eisis
  b2 feses
}

```



L'une des conséquences de toutes ces règles est que la première note rencontrée dans un fragment balisé par `\relative f` sera interprétée tout comme si elle avait été saisie en mode absolu.

Voir aussi

Glossaire musicologique : [Section “quinte” dans Glossaire](#), [Section “intervalle” dans Glossaire](#), [Section “nom des notes” dans Glossaire](#).

Manuel de notation : [\[Vérifications d’octave\]](#), page 9.

Morceaux choisis : [Section “Hauteurs” dans Morceaux choisis](#).

Référence des propriétés internes : [Section “RelativeOctaveMusic” dans Référence des propriétés internes](#).

Altérations

Note : Les nouveaux utilisateurs sont parfois déroutés par la gestion des altérations et de l’armure. Pour LilyPond, une hauteur n’est que du matériau brut ; l’armure et la clef ne feront que déterminer comment ce matériau sera retranscrit. Un simple `c` signifie tout bonnement « do naturel » quelles que soient l’armure et la clef en question. Pour plus d’information, reportez-vous au chapitre [Section “Altérations et armure” dans Manuel d’initiation](#).

Dans la notation par défaut, un *dièse* est formé en ajoutant `is` après le nom de note, un *bémol* en ajoutant `es`. Les *double-dièses* et *double-bémols* sont obtenus en ajoutant respectivement `isis` ou `eses` au nom de note. Ce sont les noms de note hollandais. Pour les autres langues, consultez [\[Nom des notes dans d’autres langues\]](#), page 7.

`ais1 aes aisis aeses`



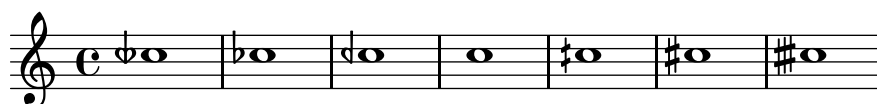
Un bécarré annule l’effet d’une altération, qu’elle soit accidentelle ou à l’armure. Cependant, dans la syntaxe des noms de note, les bécarrés ne s’écrivent pas avec un suffixe ; un simple nom de note donnera une note bécarré.

`a4 aes a2`



Les demi-bémols et demi-dièses s’écrivent en ajoutant respectivement `eh` et `ih`. Voici une série de dos altérés en hauteurs croissantes :

`ceseh1 ces ceh c cih cis cisih`



Les micro-intervalles sont aussi exportés dans le fichier MIDI.

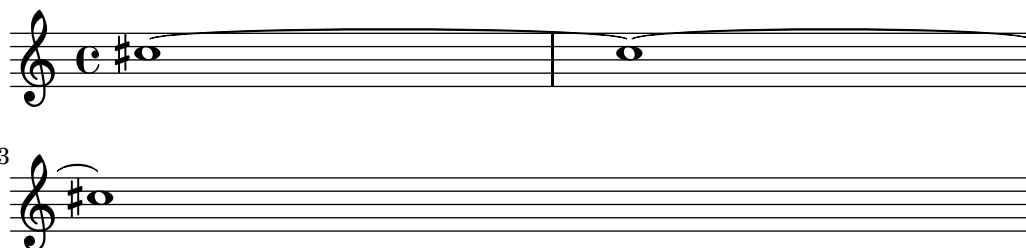
Normalement, les altérations sont imprimées automatiquement, mais il se peut que vous vouliez les imprimer manuellement. On peut forcer l’impression d’une altération, dite « de précaution », en ajoutant un point d’exclamation ! après la hauteur de note. Une altération entre parenthèses peut être obtenue en ajoutant un point d’interrogation ? après la hauteur de note. Ces signes peuvent aussi être utilisés pour imprimer des bécarrés.

```
cis cis cis! cis? c c c! c?
```



Lorsqu'une note est prolongée par une liaison de tenue, l'altération ne sera réimprimée que s'il y a un saut de ligne.

```
cis1 ~ cis ~  
\break  
cis
```

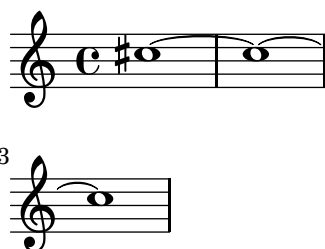


Morceaux choisis

Non répétition de l'altération après saut de ligne sur liaison de prolongation

Cet exemple illustre comment, lorsqu'une note affublée d'une altération accidentelle est prolongée, ne pas répéter cette altération après un saut de ligne.

```
\relative c'' {  
  \override Accidental #'hide-tied-accidental-after-break = ##t  
  cis1~ cis~  
  \break  
  cis  
}
```



Suppression des bécarrés superflus

En accord avec les règles standards de l'écriture musicale, on grave un bécarré avant un dièse ou un bémol si on a besoin d'annuler une altération précédente. Pour modifier ce comportement, la propriété `extraNatural` du contexte `Staff` doit se voir attribuer la valeur `##f` (faux).

```
\relative c'' {  
  aeses4 aes ais a  
  \set Staff.extraNatural = ##f  
  aeses4 aes ais a  
}
```



Voir aussi

Glossaire musicologique : Section “dièse” dans *Glossaire*, Section “bémol” dans *Glossaire*, Section “double dièse” dans *Glossaire*, Section “double bémol” dans *Glossaire*, Section “Nom des notes” dans *Glossaire*, Section “quart de ton” dans *Glossaire*.

Manuel d’initiation : Section “Altérations et armure” dans *Manuel d’initiation*.

Manuel de notation : [Altérations accidentelles automatiques], page 23, [Altérations suggérées (musica ficta)], page 417, [Nom des notes dans d’autres langues], page 7.

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Références des propriétés internes : Section “Accidental engraver” dans *Référence des propriétés internes*, Section “Accidental” dans *Référence des propriétés internes*, Section “AccidentalCautionary” dans *Référence des propriétés internes*, Section “accidental-interface” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Il n’y a pas de standard universellement accepté pour noter le bémol et demi (qui abaisse la hauteur trois quarts de ton), le symbole de LilyPond n’est donc conforme à aucun standard.

Nom des notes dans d’autres langues

Vous disposez de jeux prédéfinis de noms de note et altérations pour plusieurs autres langues. Pour les utiliser, il suffit de déclarer, en début de fichier, la langue que vous utilisez. Voici comment, par exemple, utiliser l’italien pour votre saisie :

```
\language "italiano"

\relative do' {
  do re mi sib
}
```



Les langues disponibles ainsi que les noms de note utilisés sont les suivants :

Langue	Nom des notes
nederlands	c d e f g a bes b
catalan	do re mi fa sol la sib si
deutsch	c d e f g a b h
english	c d e f g a bf b
espanol	do re mi fa sol la sib si
italiano	do re mi fa sol la sib si
norsk	c d e f g a b h
portugues	do re mi fa sol la sib si
suomi	c d e f g a b h
svenska	c d e f g a b h
vlaams	do re mi fa sol la sib si

et les suffixes d’altération correspondants :

Langue	dièse	bémol	double dièse	double bémol
--------	-------	-------	--------------	-----------------

nederlands	-is	-es	-isis	-eses
catalan	-d/-s	-b	-dd/-ss	-bb
deutsch	-is	-es	-isis	-eses
english	-s/-sharp	-f/-flat	-ss/-x/-sharpsharp	-ff/-flatflat
espanol	-s	-b	-ss/-x	-bb
italiano	-d	-b	-dd	-bb
norsk	-iss/-is	-ess/-es	-ississ/-isis	-essess/-eses
portugues	-s	-b	-ss	-bb
suomi	-is	-es	-isis	-eses
svenska	-iss	-ess	-ississ	-essess
vlaams	-k	-b	-kk	-bb

Notez qu'en hollandais, en allemand, en norvégien et en suédois, un *la* altéré de *bémol* se note *aes* et *aeses*. Ces formes sont contractées en *as* et *ases* (ou plus communément *asas*). Dans certains fichiers linguistiques, seules ces formes abrégées ont été définies.

a2 as e es a ases e eses



Certaines musiques utilisent des microtonalités, pour lesquelles les altérations sont des fractions de dièse ou bémol « normaux ». Le tableau suivant répertorie les noms de note en quart de ton, tels que définis dans plusieurs fichiers linguistiques. Les préfixes *semi-* et *sesqui-* correspondent au *demi-* et *trois demis*. À noter qu'aucune définition n'existe à ce jour pour le norvégien, le suédois, le catalan et l'espagnol.

Langue	semi-dièse	semi-bémol	sesqui-dièse	sesqui-bémol
nederlands	-ih	-eh	-isih	-eseh
deutsch	-ih	-eh	-isih	-eseh
english	-qs	-qf	-tqs	-tqf
espanol	-cs	-cb	-tcs	-tcb
italiano	-sd	-sb	-dsd	-bsb
portugues	-sqt	-bqt	-stqt	-btqt

La plupart des langues dont nous venons de parler correspondent à la musique classique occidentale au tempérament égal – le concept de *Common Practice Period* en anglais. Lily-Pond prend néanmoins en charge d'autres systèmes de notation, comme indiqué au chapitre [Section 2.10.1 \[Noms des notes et altérations non-occidentaux\]](#), page 437.

Voir aussi

Glossaire musicologique : [Section “Nom des notes” dans Glossaire](#), [Section “Common Practice Period” dans Glossaire](#).

Manuel de notation : [Section 2.10.1 \[Noms des notes et altérations non-occidentaux\]](#), page 437.

Fichiers d'initialisation : `'scm/define-note-names.scm'`.

Morceaux choisis : [Section “Hauteurs” dans Morceaux choisis](#).

1.1.2 Modification de plusieurs hauteurs

Cette partie traite de la manière de modifier les hauteurs de note.

Vérifications d'octave

Les tests d'octave rendent la correction d'erreurs d'octave plus facile dans le mode d'octave **relative** – un `,` ou un `'` oublié, ça n'arrive pas qu'aux autres !

Une note peut être suivie de *=apostrophes/virgules* pour indiquer à quelle **octave absolue** elle devrait être. Dans l'exemple suivant, le premier `d` générera un avertissement, puisqu'on attend un `d''` – intervalle inférieur à la quarte – mais qu'on obtient un `d'`. Sur la partition, l'octave sera corrigée pour donner un `d'` et la prochaine note sera calculée en fonction de ce `d'` et non de `d''`.

```
\relative c'' {
  c2 d='4 d
  e2 f
}
```



Il existe aussi une vérification d'octave qui ne produit pas de musique imprimée, ayant pour syntaxe `\octaveCheck hauteur_référence – hauteur_référence` étant spécifiée en mode absolu. Cette commande vérifie que l'intervalle entre la note qui précède et *hauteur_référence* est inférieur à la quinte comme il se doit en mode relatif. Dans le cas contraire, un message sera émis et la note précédente ne sera pas modifiée. Les notes suivantes seront, par contre, positionnées relativement à *hauteur_référence*.

```
\relative c'' {
  c2 d
  \octaveCheck c'
  e2 f
}
```



Dans les deux mesures qui suivent, les premier et troisième `\octaveCheck` échouent, mais le deuxième est concluant.

```
\relative c'' {
  c4 f g f

  c4
  \octaveCheck c'
  f
  \octaveCheck c'
  g
  \octaveCheck c'
  f
}
```



Voir aussi

Morceaux choisis : [Section “Hauteurs” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “RelativeOctaveCheck” dans *Référence des propriétés internes*](#).

Transposition

Une expression musicale peut être transposée avec `\transpose`. En voici la syntaxe :

```
\transpose note_de_départ note_d_arrivée expression_musicale
```

Cela signifie que *expression_musicale* est transposé de l'intervalle séparant *note_de_départ* et *note_d_arrivée* : toute note dont la hauteur était *note_de_départ* est changée en *note_d_arrivée* ; les autres notes seront changées selon le même intervalle. Les deux hauteurs s'expriment en octave absolue.

Note : La musique contenue dans un bloc `\transpose` est en octaves absolues, sauf à inclure dans ce même bloc une clause `\relative`.

Prenons comme exemple une pièce écrite en ré majeur. Si cette pièce est un peu trop basse pour l'interprète, elle peut être transposée en mi majeur. Vous noterez que l'armure est automatiquement modifiée.

```
\transpose d e {
  \relative c' {
    \key d \major
    d4 fis a d
  }
}
```



Regardons maintenant une partie écrite pour violon – un instrument en ut. Si cette partie doit être jouée par une clarinette en la (écrite à la tierce mineure supérieure, un do écrit donnant un la réel), la transposition suivante créera la partie appropriée.

```
\transpose a c' {
  \relative c' {
    \key c \major
    c4 d e g
  }
}
```



La présence de `\key c \major` s'explique par le fait que, bien que les notes soient effectivement transposées, l'armure ne sera imprimée que dans la mesure où elle est explicitement mentionnée.

`\transpose` fait la distinction entre les notes enharmoniques : `\transpose c cis` et `\transpose c des` transposeront la pièce un demi-ton plus haut, au détail près que la première version écrira des dièses et la deuxième des bémols.

```

music = \relative c' { c d e f }
\new Staff {
  \transpose c cis { \music }
  \transpose c des { \music }
}

```



On peut aussi utiliser `\transpose` pour entrer des notes écrites pour un instrument transpositeur. Normalement, les hauteurs dans LilyPond sont écrites en ut, c'est à dire en sons réels, mais elles peuvent être écrites dans un autre ton. Prenons l'exemple d'un morceau pour trompette en si bémol commençant sur un ré à l'oreille ; on pourrait écrire

```

musiqueEnSiBemol = { e4 ... }
\transpose c bes, \musiqueEnSiBemol

```

Pour imprimer cette musique en fa – et de ce fait produire une partie de cor au lieu d'un conducteur en notes réelles – on utilisera un deuxième `\transpose` :

```

musiqueEnSiBemol = { e4 ... }
\transpose f c' { \transpose c bes, \musiqueEnSiBemol }

```

Pour plus d'information à ce sujet, consultez [\[Instruments transpositeurs\]](#), page 22.

Morceaux choisis

Transposition et réduction du nombre d'altérations accidentelles

Cet exemple, grâce à un peu de code Scheme, donne la priorité aux enharmoniques afin de limiter le nombre d'altérations supplémentaires. La règle applicable est :

- Les altérations doubles sont supprimées
- Si dièse -> Do
- Mi dièse -> Fa
- Do bémol -> Si
- Fa bémol -> Mi

Cette façon de procéder aboutit à plus d'enharmoniques naturelles.

```

#(define (naturalize-pitch p)
  (let ((o (ly:pitch-octave p))
        (a (* 4 (ly:pitch-alteration p)))
        ;; alteration, a, in quarter tone steps,
        ;; for historical reasons
        (n (ly:pitch-notename p)))
    (cond
      ((and (> a 1) (or (eq? n 6) (eq? n 2))))
      (set! a (- a 2))
      (set! n (+ n 1)))
    ((and (< a -1) (or (eq? n 0) (eq? n 3))))
    (set! a (+ a 2))
    (set! n (- n 1))))
  (cond
    ((> a 2) (set! a (- a 4)) (set! n (+ n 1)))
    ((< a -2) (set! a (+ a 4)) (set! n (- n 1))))
  (if (< n 0) (begin (set! o (- o 1)) (set! n (+ n 7)))))

```

```

      (if (> n 6) (begin (set! o (+ o 1)) (set! n (- n 7))))
      (ly:make-pitch o n (/ a 4))))

#(define (naturalize music)
  (let ((es (ly:music-property music 'elements))
        (e (ly:music-property music 'element))
        (p (ly:music-property music 'pitch)))
    (if (pair? es)
        (ly:music-set-property!
         music 'elements
         (map (lambda (x) (naturalize x)) es)))
        (if (ly:music? e)
            (ly:music-set-property!
             music 'element
             (naturalize e)))
            (if (ly:pitch? p)
                (begin
                 (set! p (naturalize-pitch p))
                 (ly:music-set-property! music 'pitch p)))
                music)))

naturalizeMusic =
#(define-music-function (parser location m)
  (ly:music?)
  (naturalize m))

music = \relative c' { c4 d e g }

\score {
  \new Staff {
    \transpose c ais { \music }
    \naturalizeMusic \transpose c ais { \music }
    \transpose c deses { \music }
    \naturalizeMusic \transpose c deses { \music }
  }
  \layout { }
}

```



Voir aussi

Manuel de notation : [\[Octaves relatives\]](#), page 2, [\[Instruments transpositeurs\]](#), page 22, [\[Inversion\]](#), page 13, [\[Transformations modales\]](#), page 14, [\[Rétrogradation\]](#), page 13.

Morceaux choisis : [Section “Hauteurs”](#) dans *Morceaux choisis*.

Référence des propriété internes : [Section “TransposedMusic”](#) dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Si vous voulez utiliser en même temps `\transpose` et `\relative`, vous devez mettre `\transpose` en dehors de `\relative`, puisque `\relative` n'aura aucun effet sur la musique apparaissant dans un `\transpose`.

La fonction `\transpose` ne permet pas d'imprimer des altérations triples ; elle les remplacera par un « équivalent enharmonique » – par exemple ré bémol au lieu de mi triple bémol.

Inversion

Une expression musicale peut s'inverser et être transposée à l'aide de l'instruction

```
\inversion hauteur-pivot hauteur-arrivée expression_musicale
```

L'*expression_musicale* sera alors inversée, intervalle par intervalle, puis transposée de telle sorte que *hauteur-pivot* devienne *hauteur-arrivée*.

```
music = \relative c' { c d e f }
\new Staff {
  \music
  \inversion d' d' \music
  \inversion d' ees' \music
}
```



Note : Le motif à inverser doit être exprimé en hauteur absolue, à moins d'avoir été préalablement inclus explicitement dans un bloc `\relative`.

Voir aussi

Manuel de notation : [\[Transformations modales\]](#), page 14, [\[Rétrogradation\]](#), page 13, [\[Transposition\]](#), page 10.

Rétrogradation

Une expression musicale peut se renverser et se présenter sous forme rétrograde :

```
music = \relative c' { c8. ees16( fis8. a16 b8.) gis16 f8. d16 }
\new Staff {
  \music
  \retrograde \music
}
```



Problèmes connus et avertissements

Une liaison de prolongation à l'intérieur d'un bloc `\retrograde` sera de fait rompue, ce qui générera un message d'avertissement. Des liaisons de prolongation seront toutefois générées automatiquement dès lors qu'est activé le [\[Découpage automatique des notes\]](#), page 72.

Voir aussi

Manuel de notation : [\[Inversion\]](#), page 13, [\[Transformations modales\]](#), page 14, [\[Transposition\]](#), page 10.

Transformations modales

Dans une composition basée sur une gamme, un même motif est transformé à plusieurs reprises et selon des schémas différents. Il peut être *transposé* pour partir de différents points de la gamme ou bien être *inversé* à partir d'une note pivot dans la gamme. Il peut aussi être renversé pour produire une rétrogradation.

Note : Toute note qui ne ferait pas partie de la gamme en question ne sera pas transformée.

Transposition modale

Un motif peut se transposer selon une gamme donnée :

```
\modalTranspose hauteur-départ hauteur-arrivée gamme motif
```

Les notes de *motif* seront décalées à l'intérieur de la *gamme* selon leur degré, déterminé par l'intervalle entre *hauteur-départ* et *hauteur-arrivée* :

```
diatonicScale = \relative c' { c d e f g a b }
```

```
motif = \relative c' { c8 d e f g a b c }
```

```
\new Staff {
  \motif
  \modalTranspose c f \diatonicScale \motif
  \modalTranspose c b, \diatonicScale \motif
}
```



Il est ainsi possible de déterminer une gamme ascendante, quels qu'en soient l'amplitude et les différents intervalles :

```
pentatonicScale = \relative c' { ges aes bes des ees }
```

```
motif = \relative c' { ees8 des ges,4 <ges' bes,> <ges bes,> }
```

```
\new Staff {
  \motif
  \modalTranspose ges ees' \pentatonicScale \motif
}
```



L'utilisation de `\modalTranspose` avec une gamme chromatique produit les mêmes effets qu'un `\transpose`, à ceci près que les notes seront alors prédéterminées :

```
chromaticScale = \relative c' { c cis d dis e f fis g gis a ais b }
```

```
motif = \relative c' { c8 d e f g a b c }
```



```

\new Staff {
  \motif
  \transpose c f \motif
  \modalTranspose c f \chromaticScale \motif
}

```



Inversion modale

Un motif peut s'inverser selon une certaine gamme et à partir d'un pivot déterminé, puis transposé, le tout en une seule opération :

```

\modalInversion hauteur-pivot hauteur-arrivée gamme motif

```

Les notes de *motif* se retrouvent au même degré par rapport à *hauteur-pivot* dans la *gamme*, toutefois dans le sens opposé, puis décalées dans cette même *gamme* de l'intervalle séparant *hauteur-départ* et *hauteur-arrivée*.

Il est donc possible de simplement inverser à partir d'une des notes de la gamme en donnant la même valeur à *hauteur-départ* et *hauteur-arrivée* :

```

octatonicScale = \relative c' { ees f fis gis a b c d }
motif = \relative c' { c8. ees16 fis8. a16 b8. gis16 f8. d16 }

\new Staff {
  \motif
  \modalInversion fis' fis' \octatonicScale \motif
}

```



Pour permuter deux notes de la gamme, il suffit donc d'inverser à partir de l'une des notes et de transposer d'un degré de la gamme. Les deux notes spécifiées peuvent s'interpréter comme étant les bornes du pivot.

```

scale = \relative c' { c g' }
motive = \relative c' { c c g' c, }

\new Staff {
  \motive
  \modalInversion c' g' \scale \motive
}

```



L'opération conjointe d'une inversion et d'une rétrogradation produit une rétrogradation inversée :

```

octatonicScale = \relative c' { ees f fis gis a b c d }
motif = \relative c' { c8. ees16 fis8. a16 b8. gis16 f8. d16 }

\new Staff {
  \motif
  \retrograde \modalInversion c' c' \octatonicScale \motif
}

```



Voir aussi

Manuel de notation : [\[Inversion\]](#), page 13, [\[Rétrogradation\]](#), page 13, [\[Transposition\]](#), page 10.

1.1.3 Gravure des hauteurs

Nous allons voir dans cette partie comment influencer sur la gravure des hauteurs.

Clefs

La clef indique quelles lignes de la portée correspondent à quelles hauteurs. Elle peut changer au fil du morceau, comme dans l'exemple suivant qui indique le do médium dans différentes clefs :

```

\clef treble
c2 c
\clef alto
c2 c
\clef tenor
c2 c
\clef bass
c2 c

```



LilyPond met à votre disposition d'autres clefs :

```

\clef french
c2 c
\clef soprano
c2 c
\clef mezzosoprano
c2 c
\clef baritone
c2 c

\break

\clef varbaritone
c2 c
\clef subbass
c2 c

```

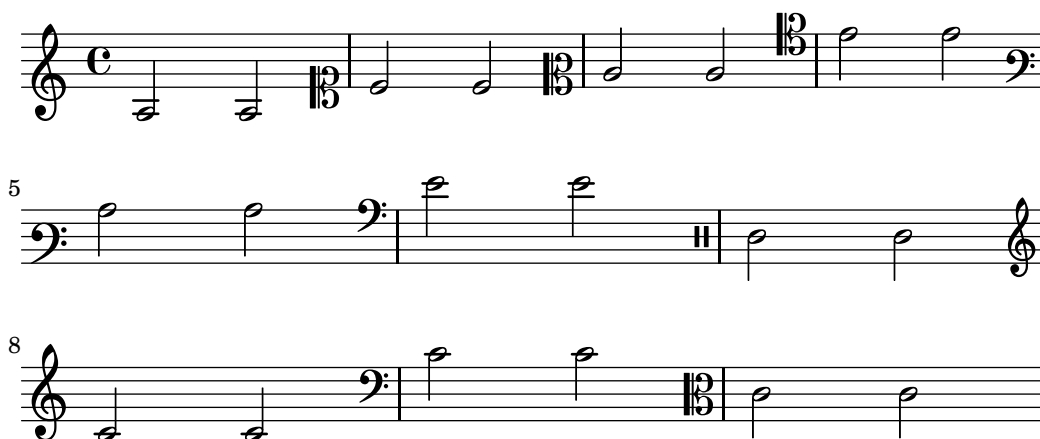
```

\clef percussion
c2 c

\break

\clef G % synonym for treble
c2 c
\clef F % synonym for bass
c2 c
\clef C % synonym for alto
c2 c

```



En ajoutant `_8` ou `^8` au nom de la clef, celle-ci est transposée à l'octave respectivement inférieure ou supérieure, et `_15` ou `^15` la transpose de deux octaves. D'autres nombres entiers peuvent être utilisés selon les besoins. L'argument *clefname* doit être mis entre guillemets lorsqu'il contient des caractères supplémentaires. Par exemple,

```

\clef treble
c2 c
\clef "treble_8"
c2 c
\clef "bass^15"
c2 c
\clef "alto_2"
c2 c
\clef "G_8"
c2 c
\clef "F^5"
c2 c

```



Il existe d'autres clefs, destinées à un usage particulier. Voir à ce sujet [Section 2.9 \[Notations anciennes\]](#), page 407, [\[Clefs anciennes\]](#), page 412, [\[Clefs grégoriennes\]](#), page 420, [\[Tablatures par défaut\]](#), page 319 et [\[Tablatures personnalisées\]](#), page 331. La nécessité de recourir à d'autres clefs dans le cas où la partition comprend des petites notes est abordée au chapitre [\[Mise en forme d'une citation\]](#), page 195 – voir les fonctions `\cueClef` et `\cueDuringWithClef`.

Morceaux choisis

Affinage des propriétés d'une clef

La commande `\clef "treble_8"` équivaut à définir `clefGlyph`, `clefPosition` – qui contrôle la position verticale de la clef – `middleCPosition` et `clefOctavation`. Une clef est imprimée lorsque l'une de ces propriétés, hormis `middleCPosition`, est modifiée. Les exemples suivant font apparaître des possibilités de réglage manuel de ces propriétés.

Modifier le glyphe, la position de la clef ou son octavation ne changera pas la position des notes ; il faut pour y parvenir modifier aussi la position du do médium. Le positionnement est relatif à la ligne médiane, un nombre positif faisant monter, chaque ligne ou interligne comptant pour 1. La valeur de `clefOctavation` devrait être de 7, -7, 15 ou -15, bien que rien n'empêche de lui affecter une autre valeur.

Lorsqu'un changement de clef intervient en même temps qu'un saut de ligne, la nouvelle clef est imprimée à la fois en fin de ligne et au début de la suivante. Vous pouvez toujours supprimer cette « clef de précaution » en affectant la valeur `end-of-line-invisible` à la propriété `explicitClefVisibility` du contexte `Staff`. Le comportement par défaut sera réactivé par `\unset Staff.explicitClefVisibility`.

Les exemples qui suivent illustrent les différentes possibilités de définir ces propriétés manuellement. Sur la première ligne, la position relative des notes par rapport aux clefs est préservée, ce qui n'est pas le cas pour la deuxième ligne.

```
\layout { ragged-right = ##t }

{
  % The default treble clef
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
  c'1
  % The baritone clef
  \set Staff.clefGlyph = #"clefs.C"
  \set Staff.clefPosition = #4
  \set Staff.middleCPosition = #4
  c'1
  % The standard choral tenor clef
  \set Staff.clefGlyph = #"clefs.G"
  \set Staff.clefPosition = #-2
  \set Staff.clefOctavation = #-7
  \set Staff.middleCPosition = #1
  c'1
  % A non-standard clef
  \set Staff.clefPosition = #0
  \set Staff.clefOctavation = #0
  \set Staff.middleCPosition = #-4
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes and clefs:

  \set Staff.clefGlyph = #"clefs.F"
```

```

\set Staff.clefPosition = #2
c'1
\set Staff.clefGlyph = #"clefs.G"
c'1
\set Staff.clefGlyph = #"clefs.C"
c'1
\set Staff.clefOctavation = #7
c'1
\set Staff.clefOctavation = #0
\set Staff.clefPosition = #0
c'1
}

% Return to the normal clef:

\set Staff.middleCPosition = #0
c'1
}

```



Voir aussi

Manuel de notation : [Section 2.9 \[Notations anciennes\]](#), page 407, [\[Clefs anciennes\]](#), page 412, [\[Clefs grégoriennes\]](#), page 420, [\[Mise en forme d’une citation\]](#), page 195, [\[Tablatures par défaut\]](#), page 319, [\[Tablatures personnalisées\]](#), page 331.

Morceaux choisis: [Section “Hauteurs”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Clef_engraver”](#) dans *Référence des propriétés internes*, [Section “Clef”](#) dans *Référence des propriétés internes*, [Section “OctavateEight”](#) dans *Référence des propriétés internes*, [Section “clef-interface”](#) dans *Référence des propriétés internes*.

Armure

Note : Les nouveaux utilisateurs sont parfois déroutés par la gestion des altérations et de l’armure. Pour LilyPond, une hauteur n’est que du matériau brut ; l’armure et la clef ne feront que déterminer comment ce matériau sera retranscrit. Un simple `c` signifie tout bonnement « do naturel » quelles que soient l’armure et la clef en question. Pour plus d’information, reportez-vous au chapitre [Section “Altérations et armure”](#) dans *Manuel d’initiation*.

L’armure indique la tonalité dans laquelle la pièce doit être jouée. Elle comprend un ensemble d’altérations (dièses ou bémols) à la clef, c’est-à-dire au début de la portée. Elle peut varier en cours de morceau.

On définit ou modifie l’armure avec la commande `\key` :

```
\key g \major
fis1
f
fis
```



```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



```
\relative c' {
  \set Staff.keySignature = #`(((0 . 6) . ,FLAT)
                                     ((0 . 5) . ,FLAT)
```

```

((0 . 3) . ,SHARP))

c4 d e fis
aes4 bes c2
}

```



Voir aussi

Glossaire musicologique : [Section “mode d’église”](#) dans *Glossaire*, [Section “scordatura”](#) dans *Glossaire*.

Manuel d’initiation : [Section “Altérations et armure”](#) dans *Manuel d’initiation*.

Morceaux choisis : [Section “Hauteurs”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “KeyChangeEvent”](#) dans *Référence des propriétés internes*, [Section “Key-engraver”](#) dans *Référence des propriétés internes*, [Section “Key-performer”](#) dans *Référence des propriétés internes*, [Section “KeyCancellation”](#) dans *Référence des propriétés internes*, [Section “KeySignature”](#) dans *Référence des propriétés internes*, [Section “key-cancellation-interface”](#) dans *Référence des propriétés internes*, [Section “key-signature-interface”](#) dans *Référence des propriétés internes*.

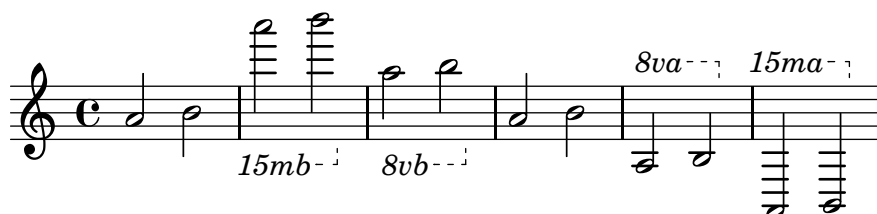
Marques d’octaviation

Les marques d’octaviation, *Ottava*, permettent d’introduire une transposition spécifique d’une octave pour la portée en cours. C’est la fonction `ottava` qui s’en charge.

```

a2 b
\ottava #-2
a2 b
\ottava #-1
a2 b
\ottava #0
a2 b
\ottava #1
a2 b
\ottava #2
a2 b

```

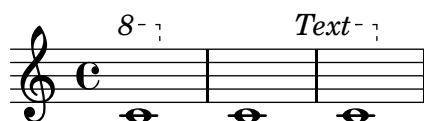


Morceaux choisis

Texte des marques d’octaviation

En interne, la fonction `\ottava` détermine les propriétés `ottavation` (par ex. en "8va" ou "8vb") et `centralCPosition`. Vous pouvez modifier le texte d’une marque d’octaviation en définissant `ottavation` après avoir fait appel à `ottava` :

```
{
  \ottava #1
  \set Staff.ottavation = #"8"
  c''1
  \ottava #0
  c'1
  \ottava #1
  \set Staff.ottavation = #"Text"
  c''1
}
```



Voir aussi

Glossaire musicologique : [Section “octaviation” dans *Glossaire*](#).

Morceaux choisis : [Section “Hauteurs” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “Ottava_spanner_engraver” dans *Référence des propriétés internes*](#), [Section “OttavaBracket” dans *Référence des propriétés internes*](#), [Section “ottava-bracket-interface” dans *Référence des propriétés internes*](#).

Instruments transpositeurs

Lorsque l’on saisit une partition d’ensemble incluant des instruments transpositeurs, certaines parties peuvent être dans une autre tonalité que la *tonalité de concert*. Il faudra en pareil cas indiquer la tonalité spécifique de ces *instruments transpositeurs*, sous peine de fichier MIDI erroné et de citations incorrectes. Pour plus de détails sur les citations, consultez le chapitre [\[Citation d’autres voix\]](#), [page 192](#).

```
\transposition hauteur
```

La hauteur donnée en argument à `\transposition` doit correspondre à la note entendue lorsqu’un *do* écrit sur la portée est joué par l’instrument transpositeur. Cette hauteur doit être mentionnée en *mode absolu*. Par exemple, lorsque vous saisissez une partition en notes réelles, toutes les voix devraient être en ut ; si un instrument joue un ton au dessus, il faudra lui ajouter un `\transposition d'`. La commande `\transposition` s’utilise **si et seulement si** les notes à saisir **ne sont pas** dans la tonalité de concert.

Voici un fragment pour violon et clarinette en si bémol (*B-flat*) pour lequel les parties respectives ont été recopiées à partir du conducteur. Les deux instruments sont à l’unisson.

```
\new GrandStaff <<
  \new Staff = "violin" {
    \relative c'' {
      \set Staff.instrumentName = #"Vln"
      \set Staff.midiInstrument = #"violin"
      % not strictly necessary, but a good reminder
      \transposition c'

      \key c \major
      g4( c8) r c r c4
    }
  }
}
```



```

\new Staff = "clarinet" {
  \relative c'' {
    \set Staff.instrumentName = \markup { Cl (B\flat) }
    \set Staff.midiInstrument = #"clarinet"
    \transposition bes

    \key d \major
    a4( d8) r d r d4
  }
}
>>

```



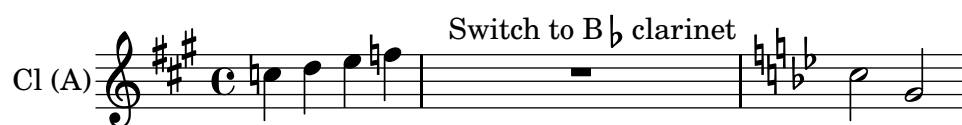
La `\transposition` peut évoluer au cours d’un morceau. Un clarinettiste peut être amené à jongler avec une clarinette en la et une autre en si bémol.

```

\set Staff.instrumentName = #"Cl (A)"
\key a \major
\transposition a
c d e f
\textLengthOn
<>^\markup { Switch to B\flat clarinet }
R1

\key bes \major
\transposition bes
c2 g

```



Voir aussi

Glossaire musicologique : [Section “tonalité de concert”](#) dans *Glossaire*, [Section “instrument transpositeur”](#) dans *Glossaire*.

Manuel de notation : [\[Citation d’autres voix\]](#), page 192, [\[Transposition\]](#), page 10.

Morceaux choisis : [Section “Hauteurs”](#) dans *Morceaux choisis*.

Altérations accidentelles automatiques

LilyPond dispose d’une fonction chargée de regrouper les règles suivant lesquelles s’impriment les altérations. Elle s’invoque de la manière suivante :

```

\new Staff <<
  \accidentalStyle "voice"
  { ... }

```

>>

La règle de gestion des altérations s'applique par défaut au contexte **Staff** en cours, exception faite des styles **piano** et **piano-cautionary** comme nous allons le voir. Cette fonction accepte un éventuel argument supplémentaire chargé de spécifier le champ d'action de la règle à suivre. À titre d'exemple, il faudra utiliser, pour que toutes les portées d'un même système – contexte **StaffGroup** – soient soumises à la même règle :

```
\accidentalStyle #'StaffGroup "voice"
```

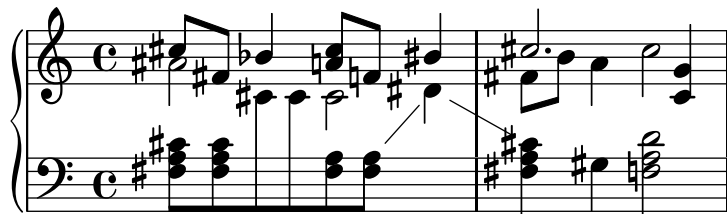
Nous vous présentons ci-après les différentes règles d'altération prises en charge. Pour les besoins de la démonstration, nous partirons de l'exemple suivant :

```
musicA = {
  <<
    \relative c' {
      cis'8 fis, bes4 <a cis>8 f bis4 |
      cis2. <c, g'>4 |
    }
  \\\
  \relative c' {
    ais'2 cis, |
    fis8 b a4 cis2 |
  }
  >>
}
```

```
musicB = {
  \clef bass
  \new Voice {
    \voiceTwo \relative c' {
      <fis, a cis>8[ <fis a cis>
      \change Staff = up
      cis' cis
      \change Staff = down
      <fis, a> <fis a>]
      \showStaffSwitch
      \change Staff = up
      dis'4 |
      \change Staff = down
      <fis, a cis>4 gis <f a d>2 |
    }
  }
}
```

```
\new PianoStaff {
  <<
    \context Staff = "up" {
      \accidentalStyle "default"
      \musicA
    }
    \context Staff = "down" {
      \accidentalStyle "default"
      \musicB
    }
  }
```

```
>>
}
```

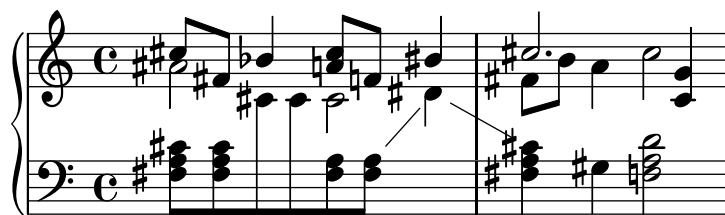


Notez bien que pour appliquer le même style aux deux portées, seules les dernières lignes de cet exemple nous intéressent.

```
\new PianoStaff {
  <<
    \context Staff = "haut" {
      %% voici la ligne à modifier en conséquence :
      \accidentalStyle #'Score "default"
      \musicA
    }
    \context Staff = "bas" {
      \musicB
    }
  >>
}
```

default

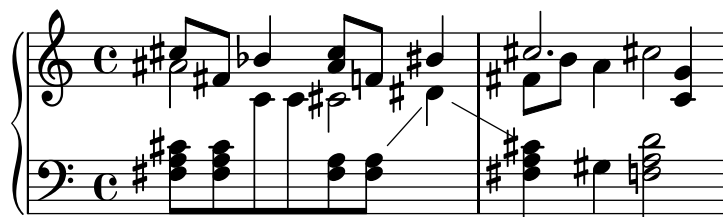
C'est la règle d'impression par défaut, qui se rapporte à l'usage en vigueur au XVIII^e siècle : les altérations accidentelles sont valables tout une mesure, et uniquement à leur propre octave. C'est la raison pour laquelle il n'y a pas de bécarré avant le *si* de la deuxième mesure, ni avant le dernier *do*.



voice

En principe, LilyPond se souvient de toutes les altérations présentes sur la portée (contexte **Staff**). Avec cette règle, cependant, les altérations sont indépendantes pour chacune des voix tout en obéissant à la règle **default**.

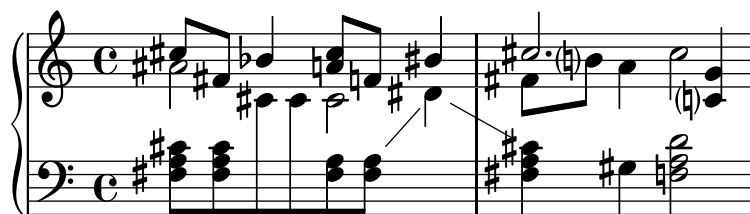
Les altérations d'une voix sont de fait ignorées dans les autres voix, ce qui peut donner lieu à un résultat malencontreux. Dans l'exemple suivant, il est difficile de dire si le deuxième *la* est dièse ou naturel. La règle **voice** n'est donc à envisager que dans le cas de voix devant être lues par des musiciens différents. S'il s'agit d'un « conducteur », ou d'une portée destinée à un seul musicien, il vaut mieux utiliser **modern** ou **modern-cautionary**.

**modern**

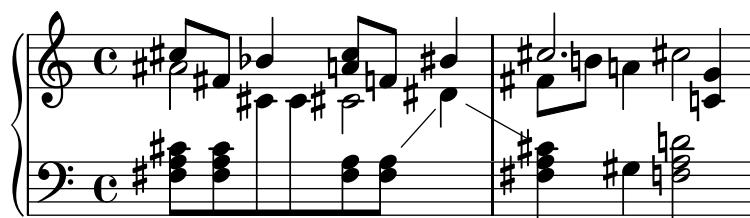
Cette règle est la plus courante au XXe siècle. Certains bécarrés ne sont pas imprimés, comme il était d'usage lorsqu'une note diésée suit cette même note flanquée d'un double dièse, ou bien un bémol un double bémol. Le style **modern** suit la même règle que le style **default**, avec deux additions afin de lever les ambiguïtés : lorsqu'une note non altérée apparaît à une octave différente, ou bien dans la mesure suivante, des bécarrés de précaution sont ajoutés. Dans l'exemple suivant, notez ainsi les deux bécarrés dans la deuxième mesure de la main droite.

**modern-cautionary**

Cette règle est équivalente à **modern**, mais les bécarrés de précaution (absents dans la règle **default**) sont imprimés de façon particulière : soit plus petit, soit (par défaut) entre parenthèses – il est possible de le définir au moyen de la propriété **cautionary-style** de l'objet **AccidentalSuggestion**.

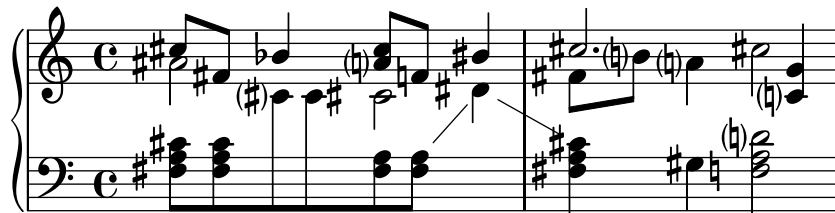
**modern-voice**

Cette règle sert aux altérations dans de la musique polyphonique destinée autant à des musiciens différents qu'à quelqu'un qui lirait l'ensemble des voix. Les altérations sont imprimées voix par voix, mais les autres voix d'un même contexte **Staff** en *tiennent compte* cette fois. C'est pourquoi le *la* de la dernière mesure est affublé d'un bémol bien qu'il y en ait déjà eu un dans la mesure précédente, et que le *ré* de la main gauche en ait un alors que le dièse qu'il avait auparavant concernait la main droite.

**modern-voice-cautionary**

Cette règle est similaire à la précédente, mais les altérations de précaution (celles que n'aurait pas ajoutées *voice*), sont imprimées de façon particulière. On retrouve

donc toutes les altérations qu'imprimerait **default**, mais certaines sont considérées comme étant « de précaution ».



piano

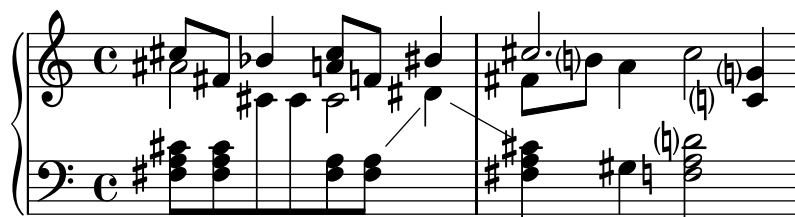
Cette règle est communément employée pour les partitions de piano au XXe siècle. Très similaire à **modern** de par son comportement, elle s'en distingue en ce que les altérations tiennent compte des autres portées du contexte **GrandStaff** ou **PianoStaff**.

Cette règle s'applique par défaut dans un **GrandStaff** et dans un **PianoStaff**.



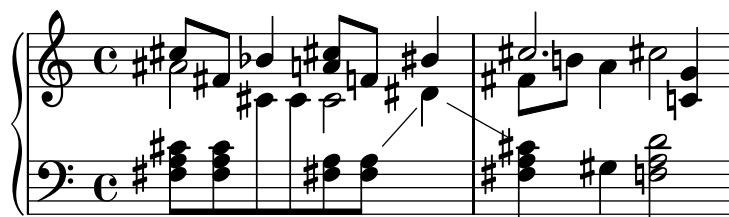
piano-cautionary

Identique au style **piano**, mais les altérations de précaution sont imprimées différemment.



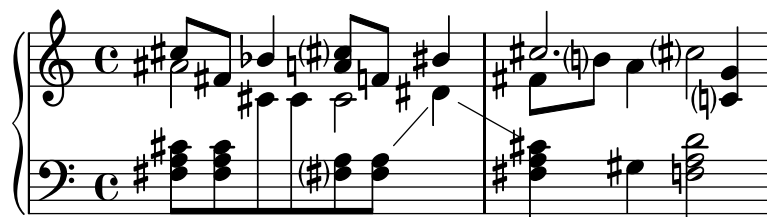
neo-modern

Cette règle suit les pratiques de la musique contemporaine : les altérations accidentelles apparaissent comme dans le style **modern**, à ceci près qu'elles sont répétées dans la même mesure – sauf si elles concernent deux notes consécutives.

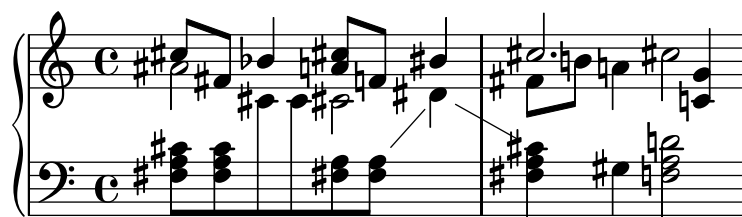


neo-modern-cautionary

Identique au style **neo-modern**, mais les altérations de précaution sont imprimées différemment.

**neo-modern-voice**

Cette règle sert aux altérations dans de la musique polyphonique destinée autant à des musiciens différents qu'à quelqu'un qui lirait l'ensemble des voix. Les altérations sont imprimées voix par voix comme avec le style **neo-modern** mais les autres voix dans le même contexte **Staff** en tiennent aussi compte.

**neo-modern-voice-cautionary**

Cette règle est identique à **neo-modern-voice**, mais les altérations de précaution sont imprimées soit entre parenthèses (par défaut), soit en plus petit.

**dodecaphonic**

Cette règle reproduit ce que certains compositeurs du début du XXe siècle ont introduit dans leur désir d'abolir la distinction entre les notes naturelles ou non. Ainsi, **chaque** note est affublée d'une altération, même si elle est naturelle.

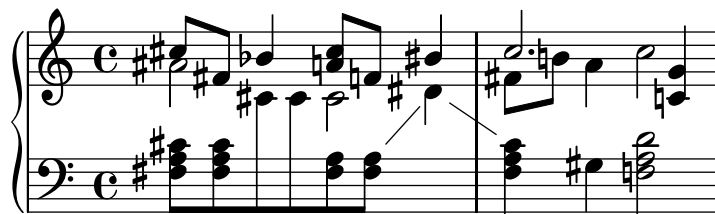
**teaching**

Cette règle est à usage pédagogique : l'impression d'une simple gamme fera apparaître une altération de précaution pour toute note altérée. Les altérations accidentelles sont imprimées selon le style **modern**, et une altération de précaution est ajoutée pour chaque dièse ou bémol à la clef – sauf dans le cas de notes consécutives.



no-reset

C'est la même règle que **default**, mais l'effet des altérations accidentelles ne cesse jamais, même dans les mesures suivantes.

**forget**

Tout le contraire de **no-reset** : l'effet des altérations cesse aussitôt ; toutes les altérations, quelque soit leur place dans la mesure, sont de ce fait imprimées en fonction de l'éventuelle armure.

**Voir aussi**

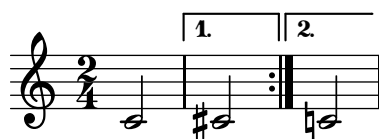
Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Accidental” dans *Référence des propriétés internes*, Section “Accidental engraver” dans *Référence des propriétés internes*, Section “GrandStaff” dans *Référence des propriétés internes* et Section “PianoStaff” dans *Référence des propriétés internes*, Section “Staff” dans *Référence des propriétés internes*, Section “AccidentalSuggestion” dans *Référence des propriétés internes*, Section “AccidentalPlacement” dans *Référence des propriétés internes*. Section “accidental-suggestion-interface” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les notes simultanées sont considérées comme des événements séquentiels. Ceci implique que, dans un accord, les altérations accidentelles seront imprimées comme si les notes de l'accord apparaissaient l'une après l'autre, en fonction de l'ordre dans lequel elles ont été saisies – ce qui peut poser problème lorsqu'au sein d'un accord certaines altérations dépendent les unes des autres. Ce problème est à résoudre manuellement, en insérant des ! et des ? après les notes concernées, tel '<f! fis!>'.

L'absence d'altération de précaution est déterminée par l'examen de la mesure précédente. Néanmoins, lorsqu'un bloc `\alternative` suit une section `\repeat volta N`, la logique voudrait que l'on regarde la dernière mesure *jouée* plutôt que la dernière *imprimée*. Dans l'exemple qui suit, vous conviendrez que le do de la seconde alternative ne nécessite pas son bécarre.

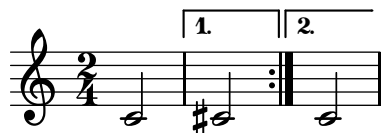


L'astuce suivante, qui définit temporairement le recours au style **forget**, permet d'obtenir quelque chose de présentable.

```

forget = #(define-music-function (parser location music) (ly:music?) #{
  \accidentalStyle "forget"
  #music
  \accidentalStyle "modern"
#})
{
  \accidentalStyle "modern"
  \time 2/4
  \repeat volta 2 {
    c'2
  }
  \alternative {
    cis'
    \forget c'
  }
}

```



Ambitus

L'*ambitus* est l'amplitude des hauteurs d'une voix donnée dans une partition. Ce terme peut aussi désigner la tessiture qu'un instrument est capable d'atteindre. Souvent, cet ambitus est imprimé au début des partitions vocales, afin que les exécutants puissent voir au premier coup d'œil s'ils sont en mesure de tenir la partie en question.

Pour exprimer l'ambitus d'une pièce, on indique avant la clef deux têtes de note représentant la hauteur la plus basse et la plus haute. Les éventuelles altérations accidentelles seront automatiquement ajoutées.

```

\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\relative c'' {
  aes c e2
  cis,1
}

```



Morceaux choisis

Un ambitus par voix

L'*ambitus* peut être individualisé par voix. Il faut en pareil cas éviter qu'ils se chevauchent.


```

\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus #'X-offset = #2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
}
>>

```



Ambitus sur plusieurs voix

Si plusieurs voix se trouvent sur une même portée, on peut attribuer le graveur `Ambitus_engraver` au contexte `Staff` afin d'obtenir l'ambitus de toutes les voix cumulées, non d'une seule des voix actives.

```

\new Staff \with {
  \consists "Ambitus_engraver"
}
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
}
>>

```



Réglage de l'affichage d'un ambitus

L'affichage d'un *ambitus* peut s'affiner pour répondre à vos préférences en matière d'esthétique.

```

\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\new Staff {
  \time 2/4
  % Default setting
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine #'gap = #0
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine #'gap = #1
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine #'gap = #1.5
  c'4 g''
}

```



Voir aussi

Glossaire musicologique : [Section “ambitus”](#) dans *Glossaire*.

Morceaux choisis : [Section “Hauteurs”](#) dans *Morceaux choisis*.

Référence des propriétés internes : Section “Ambitus_engraver” dans *Référence des propriétés internes*, Section “Voice” dans *Référence des propriétés internes*, Section “Staff” dans *Référence des propriétés internes*, Section “Ambitus” dans *Référence des propriétés internes*, Section “AmbitusAccidental” dans *Référence des propriétés internes*, Section “AmbitusLine” dans *Référence des propriétés internes*, Section “AmbitusNoteHead” dans *Référence des propriétés internes*, Section “ambitus-interface” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

LilyPond ne gère pas les collisions entre plusieurs ambitus présents sur une même portée.

1.1.4 Têtes de note

Nous allons voir dans ce chapitre comment modifier l’aspect des têtes de note.

Têtes de note spécifiques

L’apparence des têtes de note peut évoluer au cours de la partition :

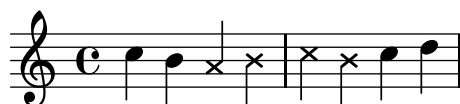
```
c4 b
\override NoteHead #'style = #'cross
c4 b
\revert NoteHead #'style
a b
\override NoteHead #'style = #'harmonic
a b
\revert NoteHead #'style
c4 d e f
```



Pour une liste exhaustive des styles de tête de note, consultez [Section A.9 \[Styles de tête de note\]](#), page 641.

Certains instruments utilisent des têtes de note différentes à des fins spécifiques – des croix (style `cross`) pour le *parlato* des chanteurs ou les notes étouffées des guitares :

```
c4 b
\xNotesOn
a b c4 b
\xNotesOff
c4 d
```



Cette commande opère aussi bien sur des notes isolées qu’au sein d’un accord, dans une portée traditionnelle ou dans un contexte de tablature :

```
c4 b
\xNote { e f }
c b < g \xNote c f > b
```



Vous pouvez utiliser, en lieu et place de `\xNote`, `\xNotesOn` et `\xNotesOff`, les commandes `\deadNote`, `\deadNotesOn` et `\deadNotesOff`.

Il existe un raccourci pour les notes en losange qui indiquent les harmoniques des cordes :

```
<c f\harmonic>2 <d a'\harmonic>4 <c g'\harmonic>
```



Commandes prédéfinies

`\harmonic`, `\xNotesOn`, `\xNotesOff`, `\xNote`.

Voir aussi

Manuel de notation : [Section A.9 \[Styles de tête de note\]](#), page 641, [\[Notes en accords\]](#), page 149, [\[Indication des harmoniques et notes étouffées\]](#), page 359.

Morceaux choisis : [Section “Hauteurs”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “note-event”](#) dans *Référence des propriétés internes*, [Section “Note_heads_engraver”](#) dans *Référence des propriétés internes*, [Section “Ledger_line_engraver”](#) dans *Référence des propriétés internes*, [Section “NoteHead”](#) dans *Référence des propriétés internes*. [Section “LedgerLineSpanner”](#) dans *Référence des propriétés internes*, [Section “note-head-interface”](#) dans *Référence des propriétés internes*, [Section “ledger-line-spanner-interface”](#) dans *Référence des propriétés internes*.

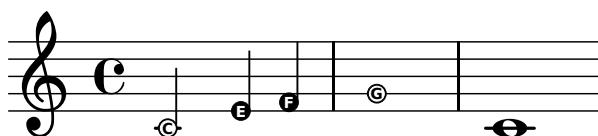
Têtes de note avec nom de note

Les notes « easy play » comportent le nom de la note à l'intérieur de la tête. On l'utilise dans des partitions pour débutants. L'impression doit être de plus grande taille, afin que les lettres soient lisibles. Voir à ce propos [Section 4.2.2 \[Définition de la taille de portée\]](#), page 509.

```

#(set-global-staff-size 26)
\relative c' {
  \easyHeadsOn
  c2 e4 f
  g1
  \easyHeadsOff
  c,1
}

```



Commandes prédéfinies

`\easyHeadsOn`, `\easyHeadsOff`.

Morceaux choisis

Easy play – chiffre en lieu et place des lettres

En mode « easy play », les têtes de note utilisent la propriété `note-names` attachée à l'objet `NoteHead` pour déterminer ce qui apparaîtra dans la tête. Intervenir sur cette propriété permet d'imprimer un chiffre correspondant au degré dans la gamme.

La création d'un graveur dédié permet de traiter toutes les notes.

```

#(define Ez_numbers_engraver
  (make-engraver
    (acknowledgers
      ((note-head-interface engraver grob source-engraver)
        (let* ((context (ly:translator-context engraver))
              (tonic-pitch (ly:context-property context 'tonic))
              (tonic-name (ly:pitch-notename tonic-pitch))
              (grob-pitch
                (ly:event-property (event-cause grob) 'pitch))
              (grob-name (ly:pitch-notename grob-pitch))
              (delta (modulo (- grob-name tonic-name) 7))
              (note-names
                (make-vector 7 (number->string (1+ delta))))))
          (ly:grob-set-property! grob 'note-names note-names))))))

#(set-global-staff-size 26)

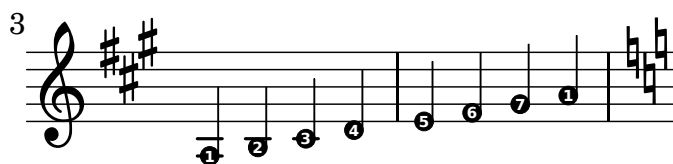
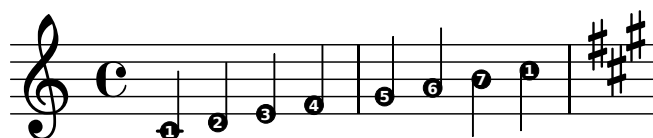
\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

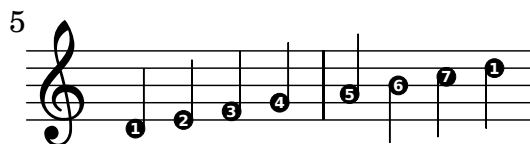
\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break

  \key a \major
  a,4 b cis d
  e4 fis gis a \break

  \key d \dorian
  d,4 e f g
  a4 b c d
}

```





Voir aussi

Manuel de notation : [Section 4.2.2 \[Définition de la taille de portée\]](#), page 509.

Morceaux choisis : [Section “Hauteurs”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “note-event”](#) dans *Référence des propriétés internes*, [Section “Note_heads_engraver”](#) dans *Référence des propriétés internes*, [Section “Note-Head”](#) dans *Référence des propriétés internes*, [Section “note-head-interface”](#) dans *Référence des propriétés internes*.

Têtes de note à forme variable

En notation profilée, le profil d’une tête de note correspond à la fonction harmonique de cette note dans la gamme. Ce style de notation était très en vogue dans les recueils de chansons américains du XIXe siècle. Voici comment procéder :

```
\aikenHeads
c, d e f g2 a b1 c \break
\sacredHarpHeads
c,4 d e f g2 a b1 c \break
\southernHarmonyHeads
c,4 d e f g2 a b1 c \break
\funkHeads
c,4 d e f g2 a b1 c \break
\walkerHeads
c,4 d e f g2 a b1 c \break
```



Les profils sont déterminés par la hauteur dans la gamme, le premier degré étant défini par la commande `\key`. Pour une tonalité mineure, les degrés sont déterminés par rapport au relatif majeur :

```

\key a \minor
\aikenHeads
a b c d e2 f g1 a \break
\aikenHeadsMinor
a,4 b c d e2 f g1 a \break
\sacredHarpHeadsMinor
a,2 b c d \break
\southernHarmonyHeadsMinor
a2 b c d \break
\funkHeadsMinor
a2 b c d \break
\walkerHeadsMinor
a2 b c d \break

```



Commandes prédéfinies

\aikenHeads, \aikenHeadsMinor, \funkHeads, \funkHeadsMinor, \sacredHarpHeads.
\sacredHarpHeadsMinor, \southernHarmonyHeads, \southernHarmonyHeadsMinor,
\walkerHeads, \walkerHeadsMinor.

Morceaux choisis

Profilage des notes selon leur degré dans la gamme

La propriété `shapeNoteStyles` permet d'affecter un profil particulier à chaque degré de la gamme – à partir de l'armure ou de la propriété `tonic`. Ses valeurs sont constituées d'une liste de symboles, qu'il s'agisse de formes géométriques (`triangle`, `cross` ou `xcircle`) ou basés sur la tradition des graveurs américains (avec quelques noms de note latins).

LilyPond dispose de deux raccourcis, `\aikenHeads` et `\sacredHarpHeads`, permettant de reproduire d’anciens recueils de chansons américaines.

L’exemple suivant montre plusieurs manières de profiler les têtes de note, ainsi que la capacité de transposer tout en respectant la fonction harmonique de chaque note dans la gamme.

```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

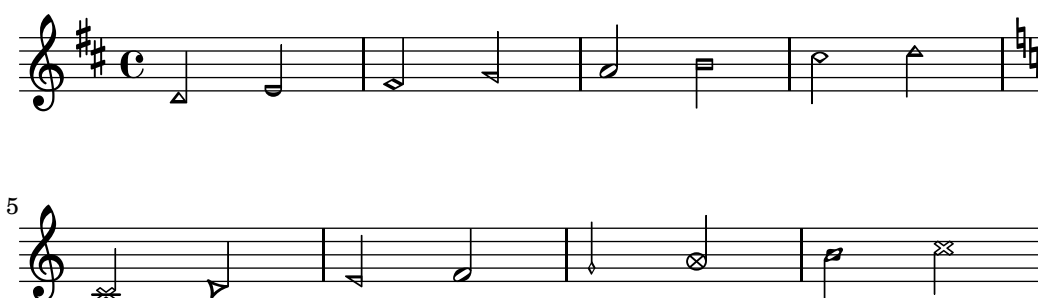
\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = #'#(do re mi fa
                          #f la ti)

    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = #'#(cross triangle fa #f
                          mensural xcircle diamond)

    \fragment
  }
}
```



Pour une liste exhaustive des styles de tête de note, consultez [Section A.9 \[Styles de tête de note\]](#), page 641.

Voir aussi

Manuel de notation : [Section A.9 \[Styles de tête de note\]](#), page 641.

Morceaux choisis : [Section “Hauteurs”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “note-event”](#) dans *Référence des propriétés internes*, [Section “Note_heads_engraver”](#) dans *Référence des propriétés internes*, [Section “Note-Head”](#) dans *Référence des propriétés internes*, [Section “note-head-interface”](#) dans *Référence des propriétés internes*.

Improvisation

L'improvisation peut quelquefois s'indiquer à l'aide de notes de forme allongée (*slash*). L'interprète jouera alors les notes qu'il veut, en respectant toutefois le rythme affiché. Ces têtes de notes sont créées ainsi :

```
\new Voice \with {
  \consists "Pitch_squash_engraver"
} {
  e8 e g a a16( bes) a8 g
  \improvisationOn
  e8 ~
  e2 ~ e8 f4 f8 ~
  f2
  \improvisationOff
  a16( bes) a8 g e
}
```



Commandes prédéfinies

\improvisationOn, \improvisationOff.

Voir aussi

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Pitch_squash_engraver” dans *Référence des propriétés internes*, Section “Voice” dans *Référence des propriétés internes*, Section “Rhythmic-Staff” dans *Référence des propriétés internes*.

1.2 Rythme



Cette section traite du rythme : durées, silences, barres de ligature et de mesure.

1.2.1 Écriture du rythme

Durées

Dans les modes de notes, d'accords et de paroles, les durées sont écrites avec des chiffres et des points : les durées sont indiquées par leur valeur fractionnaire par rapport à la durée d'une ronde. Une noire, par exemple, qui équivaut à un 1/4 de ronde – *quarter note* en anglais – s'écrit 4, alors qu'une blanche – *half-note*, 1/2 ronde – s'écrit 2. Pour des notes plus longues qu'une ronde, vous devrez utiliser les commandes `\longa` pour une longue, et `\breve` pour une brève, aussi appelée carrée. Des durées plus courtes que la quintuple croche – 1/128 de ronde – sont possibles, à condition de les ligaturer.

```
\time 8/1
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c128 c128
```



Voici ces mêmes durées sans la fonction de ligature automatique.

```
\time 8/1
\autoBeamOff
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c128 c128
```



Une note dont la durée est de quatre brèves s'obtient par la commande `\maxima`. Celle-ci n'est toutefois disponible que dans le cadre de la notation ancienne. Pour plus de détails, voir [Section 2.9 \[Notations anciennes\], page 407](#).

Si la durée d'une note n'est pas précisée, elle est alors assimilée à la durée de la note précédente. La valeur par défaut pour la première note est la noire (4).

a a a2 a a4 a a1 a



Pour obtenir des notes pointées, ajoutez simplement un point (.) au chiffre. Les notes doublement pointées sont créées de la même façon.

a4 b c4. b8 a4. b4.. c8.



Certaines durées ne peuvent s'obtenir à partir de chiffres et de points, mais uniquement en « liant » deux ou plusieurs notes entre elles. Voir [Liaisons de prolongation], page 47 à ce sujet.

Quant à la manière de spécifier la durée des syllabes ou bien d'aligner des paroles par rapport aux notes, reportez vous au chapitre Section 2.1 [Musique vocale], page 238.

Espacer les notes selon leur durée relative est tout à fait possible. Pour plus de détails à ce sujet et sur les autres réglages propres à cette forme de notation, reportez vous à Section 4.5.5 [Notation proportionnelle], page 538.

Les points sont normalement haussés pour éviter les lignes de portée, sauf dans certaines polyphonies. Des commandes prédéfinies permettent de manuellement forcer une orientation particulière, comme indiqué au chapitre Section 5.4.2 [Direction et positionnement], page 579.

Commandes prédéfinies

`\autoBeamOn`, `\autoBeamOff`, `\dotsUp`, `\dotsDown`, `\dotsNeutral`.

Morceaux choisis

Brève alternative, avec deux barres verticales

Voici comment obtenir une brève – aussi appelée note carée – flanquée de deux barres verticales, au lieu d'une comme habituellement.

```
\relative c' ' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead #'style = #'altdefault
  b\breve
  \revert Staff.NoteHead #'style
  a\breve
}
```



Spécification du nombre de points d'augmentation d'une note

Voici comment modifier le nombre de points d'augmentation affectés à une note en particulier.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots #'dot-count = #4
  c4.. a16 r2 |
}
```

```

\override Dots #'dot-count = #0
c4.. a16 r2 |
\revert Dots #'dot-count
c4.. a16 r2 |
}

```



Voir aussi

Glossaire musicologique : Section “breve” dans *Glossaire*, Section “longa” dans *Glossaire*, Section “maxima” dans *Glossaire*, Section “valeur des notes” dans *Glossaire*, Section “Noms de durée (notes et silences)” dans *Glossaire*.

Manuel de notation : [Barres de ligature automatiques], page 75, [Liaisons de prolongation], page 47, [Hampes], page 207, Section 1.2.1 [Écriture du rythme], page 40, Section 1.2.2 [Écriture des silences], page 51, Section 2.1 [Musique vocale], page 238, Section 2.9 [Notations anciennes], page 407, Section 4.5.5 [Notation proportionnelle], page 538.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Dots” dans *Référence des propriétés internes*, Section “DotColumn” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Il n'existe pas à proprement parler de limite (inférieure ou supérieure) en terme de durée d'un silence. Cependant, le nombre de glyphes disponibles ne couvre que les silences allant du trente-deuxième de soupir à la maxime (valant huit pauses).

Nolets

Les nolets – triolets, quintolets, etc. – sont obtenus en multipliant toutes les durées d'une expression musicale par une fraction.

```

\times fraction { expression_musicale }

```

La durée de l'*expression_musicale* sera multipliée par la fraction. Le dénominateur de cette fraction sera imprimé au-dessus ou au-dessous des notes, parfois avec un crochet. Le nolet le plus courant est le triolet, dans lequel trois notes ont la durée de deux, et où les notes durent donc deux tiers de leur valeur écrite.

```

a2 \times 2/3 { b4 b b }
c4 c \times 2/3 { b4 a g }

```



Des commandes prédéfinies permettent de déroger au positionnement automatique du crochet en surplomb ou au-dessous des notes – voir le chapitre [Section 5.4.2 \[Direction et positionnement\]](#), page 579.

Les nolets peuvent être imbriqués ; par exemple,

```
\autoBeamOff
c4 \times 4/5 { f8 e f \times 2/3 { e[ f g] } } f4 |
```



Lorsque, dans une imbrication, les nolets débutent au même instant, il vous faut recourir à la commande `\tweak`.

Vous pouvez interférer sur la durée des notes sans imprimer de crochet, comme indiqué au chapitre [\[Changement d'échelle des durées\]](#), page 46.

Commandes prédéfinies

`\tupletUp`, `\tupletDown`, `\tupletNeutral`.

Morceaux choisis

Plusieurs triolets avec une seule commande `\times`

La propriété `tupletSpannerDuration` spécifie la longueur voulue de chaque crochet. Avec elle, vous pouvez faire plusieurs nolets en ne tapant `\times` qu'une fois, ce qui évite une longue saisie.

Dans l'exemple suivant, deux triolets sont imprimés avec une seule fonction `\times`.

Pour plus d'information sur `make-moment`, voir la section appropriée du manuel de notation.

```
\relative c' {
  \time 2/4
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```



Modifier l'apparence du chiffre de nolet

L'apparence du chiffre est déterminée par la propriété `text` dans `TupletNumber`. La valeur par défaut imprime seulement le dénominateur, mais si elle est définie par la fonction `tuplet-number::calc-fraction-text`, la fraction entière `num:den` sera imprimée à la place.

```
\relative c' {
  \times 2/3 { c8 c c }
  \times 2/3 { c8 c c }
  \override TupletNumber #'text = #tuplet-number::calc-fraction-text
  \times 2/3 { c8 c c }
  \override TupletNumber #'stencil = ##f
  \times 2/3 { c8 c c }
}
```



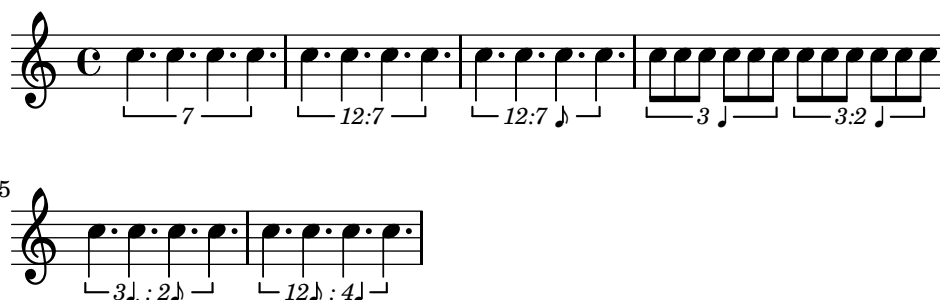
Nolets au chiffrage inhabituel

LilyPond sait aussi gérer des nolets dont le chiffrage imprimé ne correspond pas exactement à la fraction de mesure à laquelle ils se réfèrent, tout comme ceux auxquels une valeur de note vient en complément au chiffre.

```
\relative c' {
  \once \override TupletNumber #'text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \times 2/3 { c4. c4. c4. c4. }
  \once \override TupletNumber #'text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \times 2/3 { c4. c4. c4. c4. }
  \once \override TupletNumber #'text =
    #(tuplet-number::append-note-wrapper
      (tuplet-number::non-default-tuplet-fraction-text 12 7) "8")
  \times 2/3 { c4. c4. c4. c4. }

  \once \override TupletNumber #'text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-denominator-text "4")
  \times 2/3 { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber #'text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-fraction-text "4")
  \times 2/3 { c8 c8 c8 c8 c8 c8 }

  \once \override TupletNumber #'text =
    #(tuplet-number::fraction-with-notes "4." "8")
  \times 2/3 { c4. c4. c4. c4. }
  \once \override TupletNumber #'text =
    #(tuplet-number::non-default-fraction-with-notes 12 "8" 4 "4")
  \times 2/3 { c4. c4. c4. c4. }
}
```



Contrôle de l'impression des crochets de nolet

Selon la tradition, les crochets indicateurs de nolet sont toujours imprimés, sauf dans le cas où ils seraient de la même longueur qu'une ligature. LilyPond permet, au travers de la propriété '**bracket-visibility**', de contrôler précisément leur affichage : déterminée à **#t**, ils seront toujours imprimés ; **#f** permet de ne jamais les imprimer, et **#'if-no-beam** les imprimera en l'absence de ligature.

```
music = \relative c' {
  \times 2/3 { c16[ d e ] f8}
  \times 2/3 { c8 d e }
  \times 2/3 { c4 d e }
```

```

}

\new Voice {
  \relative c' {
    << \music s4^"default" >>
    \override TupletBracket #'bracket-visibility = #'if-no-beam
    << \music s4^"'if-no-beam" >>
    \override TupletBracket #'bracket-visibility = ##t
    << \music s4^"#t" >>
    \override TupletBracket #'bracket-visibility = ##f
    << \music s4^"#f" >>
  }
}

```



Saut de ligne au milieu d'un nolet avec ligature

Cet exemple peu académique démontre comment il est possible d'insérer un saut de ligne dans un nolet portant une ligature. Ces ligatures doivent toutefois être explicites.

```

\layout {
  \context {
    \Voice
    % Permit line breaks within tuplets
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks
    \override Beam #'breakable = ##t
  }
}

\relative c'' {
  a8
  \repeat unfold 5 { \times 2/3 { c[ b a] } }
  % Insert a manual line break within a tuplet
  \times 2/3 { c[ b \bar "" \break a] }
  \repeat unfold 5 { \times 2/3 { c[ b a] } }
  c8
}

```

}



Voir aussi

Glossaire musicologique : Section “triolet” dans *Glossaire*, Section “nolet” dans *Glossaire*, Section “polymétrie” dans *Glossaire*.

Manuel d’initiation : Section “Méthodes de retouche” dans *Manuel d’initiation*.

Manuel de notation : [Gestion du temps], page 108, [Changement d’échelle des durées], page 46, Section 5.3.4 [La commande d’affinage (tweak)], page 573, [Notation polymétrique], page 69.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Références des propriétés internes : Section “TupletBracket” dans *Référence des propriétés internes*, Section “TupletNumber” dans *Référence des propriétés internes*, Section “TimeScaled-Music” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Lorsqu’un nolet en début de portée est précédé d’une note d’ornement, celle-ci doit se placer avant la commande `\times` pour éviter toute erreur. Ailleurs dans la partition, les notes d’ornement peuvent se placer au sein de l’expression constituant le nolet.

Lorsqu’un nolet en début de mouvement est combiné avec une indication de tempo – grâce à la commande `\tempo` –, l’expression contenant le nolet doit impérativement faire partie d’un bloc `\new Voice` comme indiqué au chapitre Section “Les voix contiennent la musique” dans *Manuel d’initiation*.

Changement d’échelle des durées

La durée des notes, silences ou accords peut se modifier en lui adjoignant une fraction N/D , donnant « $*N/D$ » – ou « $*N$ » si $D=1$. Ceci ne modifiera en rien l’apparence des notes ou silences produits, mais affectera le positionnement de l’objet dans la mesure, ainsi que le rendu MIDI. Cette fraction peut elle-même être multipliée, ce qui donne quelque chose du style $*M*N/D$. Ce facteur d’échelonnement est partie intégrante de la durée : en l’absence de durée explicite à la note suivante, cette durée échelonnée est considérée comme valeur par défaut.

Dans l’exemple suivant, les trois premières notes prennent exactement deux temps, mais aucun triolet n’est imprimé.

```
\time 2/4
% Alter durations to triplets
a4*2/3 gis a
% Normal durations
a4 a
% Double the duration of chord
<a d>4*2
% Duration of quarter, appears like sixteenth
```


Les liaisons de tenue sont utilisées soit lorsque la note dépasse de la mesure, soit quand les points ne suffisent pas à donner la bonne durée. Lorsque l'on utilise ces liaisons, les valeurs rythmiques les plus longues doivent s'aligner sur les subdivisions de la mesure, comme ici :

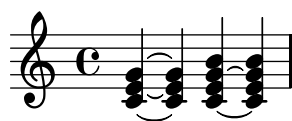
```
\relative {
  r8^"oui" c8 ~ c2 r4 |
  r8^"non" c2 ~ c8 r4
}
```



Lorsque l'on doit lier de nombreuses notes sur plusieurs mesures, il devient plus facile d'avoir recours à la division automatique des notes – voir [\[Découpage automatique des notes\]](#), page 72. Ce procédé divise automatiquement les notes trop longues, et les lie par-delà les barres de mesure.

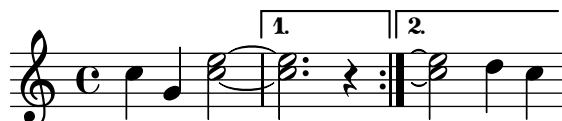
Quand une liaison de tenue se trouve entre deux accords, toutes les notes de même hauteur entre ces deux accords sont reliées. S'il n'y en a aucune, aucune liaison n'est créée. Il est également possible de lier partiellement deux accords, en mettant les liaisons à l'intérieur des accords.

```
<c e g> ~ <c e g>
<c~ e g~ b> <c e g b>
```



Lorsqu'une mesure « de seconde fois » après une reprise commence sur une note liée, la liaison doit être répétée, comme ici :

```
\repeat volta 2 { c g <c e>2 ~ }
\alternative {
  % First alternative: following note is tied normally
  { <c e>2. r4 }
  % Second alternative: following note has a repeated tie
  { <c e>2\repeatTie d4 c } }
```



Les liaisons « Laissez vibrer » (*L.v.*) sont utilisées pour le piano, la harpe et certains instruments de percussion. Elles indiquent à l'instrumentiste de laisser sonner la note ou l'accord au lieu de l'étouffer. Elles s'indiquent de la manière suivante :

```
<c f g>1\laissezVibrer
```



Le positionnement vertical des liaisons de prolongation peut être affiné – voir à ce sujet les « commandes prédéfinies » et, pour de plus amples détails, [Section 5.4.2 \[Direction et positionnement\]](#), page 579.

Les liaisons de prolongation peuvent s'imprimer sous la forme de ligne continue, discontinue ou pointillée.

```
\tieDotted
c2 ~ c
\tieDashed
c2 ~ c
\tieHalfDashed
c2 ~ c
\tieHalfSolid
c2 ~ c
\tieSolid
c2 ~ c
```



Il est même possible d'en personnaliser l'allure :

```
\tieDashPattern #0.3 #0.75
c2 ~ c
\tieDashPattern #0.7 #1.5
c2 ~ c
\tieSolid
c2 ~ c
```



Que ce soit pour une tenue ou un phrasé, le motif d'une ligne discontinue formant une liaison se définit de la même manière. Pour de plus amples détails à ce sujet, reportez vous au chapitre [\[Liaisons d'articulation\]](#), page 119.

Dans le cas où une liaison est recouverte par d'autres éléments de la portée, une adaptation des propriétés *whiteout* et *layer* permet d'obtenir une meilleure lisibilité.

```
\override Tie #'layer = #-2
\override Staff.TimeSignature #'layer = #-1
\override Staff.KeySignature #'layer = #-1
\override Staff.TimeSignature #'whiteout = ##t
\override Staff.KeySignature #'whiteout = ##t
b2 b~
\time 3/4
\key a \major
b r4
```



Commandes prédéfinies

`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieDashPattern`,
`\tieHalfDashed`, `\tieHalfSolid`, `\tieSolid`.

Morceaux choisis

Liaison de tenue et arpège

Les liaisons de tenue servent parfois à rendre un accord arpégé. Dans ce cas, les notes liées ne sont pas toutes consécutives. Il faut alors assigner à la propriété `tieWaitForNote` la valeur `#t` (*true* pour « vrai »). Cette même méthode peut servir, par exemple, à lier un trémolo à un accord.

```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```



Dessin à main levée de liaisons de tenue

Il est possible de graver manuellement les liaisons de tenue, en modifiant la propriété `tie-configuration`. Pour chaque paire, le premier nombre indique la distance à la portée, en espaces de portée, et le second la direction (1 pour haut, -1 pour bas).

```
\relative c' {
  <c e g>2~ <c e g>
  \override TieColumn #'tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
  <c e g>2~ <c e g>
}
```



Voir aussi

Glossaire musicologique : Section “liaison de tenue” dans *Glossaire*, Section “laissez vibrer” dans *Glossaire*.

Manuel de notation : [Liaisons d’articulation], page 119, [Découpage automatique des notes], page 72.

Morceaux choisis : Section “Signes d’interprétation” dans *Morceaux choisis*, Section “Rythme” dans *Morceaux choisis*.

Référence des propriétés internes : Section “LaissezVibrerTie” dans *Référence des propriétés internes*, Section “LaissezVibrerTieColumn” dans *Référence des propriétés internes*, Section “TieColumn” dans *Référence des propriétés internes*, Section “Tie” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Un changement de portée, lorsqu'une liaison de tenue est active, ne peut produire une liaison oblique.

Un changement de clé ou d'octave pendant une liaison de tenue produit un résultat indéfini. Dans ces cas là, il est préférable d'utiliser un *legato*.

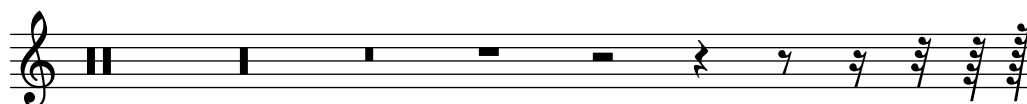
1.2.2 Écriture des silences

On saisit les silences dans une expression musicale tout comme les notes.

Silences

Les silences sont écrits comme des notes avec le nom de note *r* – premier caractère du mot *rest*. Les durées supérieures à la pause s'indiquent à l'aide de commandes prédéfinies :

```
\new Staff {
  % These two lines are just to prettify this example
  \time 16/1
  \override Staff.TimeSignature #'stencil = ##f
  % Print a maxima rest, equal to four breves
  r\maxima
  % Print a longa rest, equal to two breves
  r\longa
  % Print a breve rest
  r\breve
  r1 r2 r4 r8 r16 r32 r64 r128
}
```



Les pauses d'une mesure complète, qui sont placées au centre de la mesure, doivent être entrées comme des mesures de silence. Elles peuvent être utilisées pour une seule mesure comme pour plusieurs, et leur utilisation est expliquée à la rubrique [\[Silences valant une mesure\]](#), page 54.

Pour spécifier explicitement la position verticale d'un silence, écrivez une note suivie de `\rest`. Un silence de même durée sera placé à la position où serait imprimée la note. Cela rend plus facile la mise en place de musique polyphonique, puisque le formateur automatique de collision des silences laissera ces silences tranquilles.

```
a4\rest d4\rest
```



Morceaux choisis

Styles de silences

Les silences peuvent être gravés selon différents styles.

```
\layout {
  indent = 0
  \context {
    \Staff
    \remove "Time_signature_engraver"
```

```

}
}

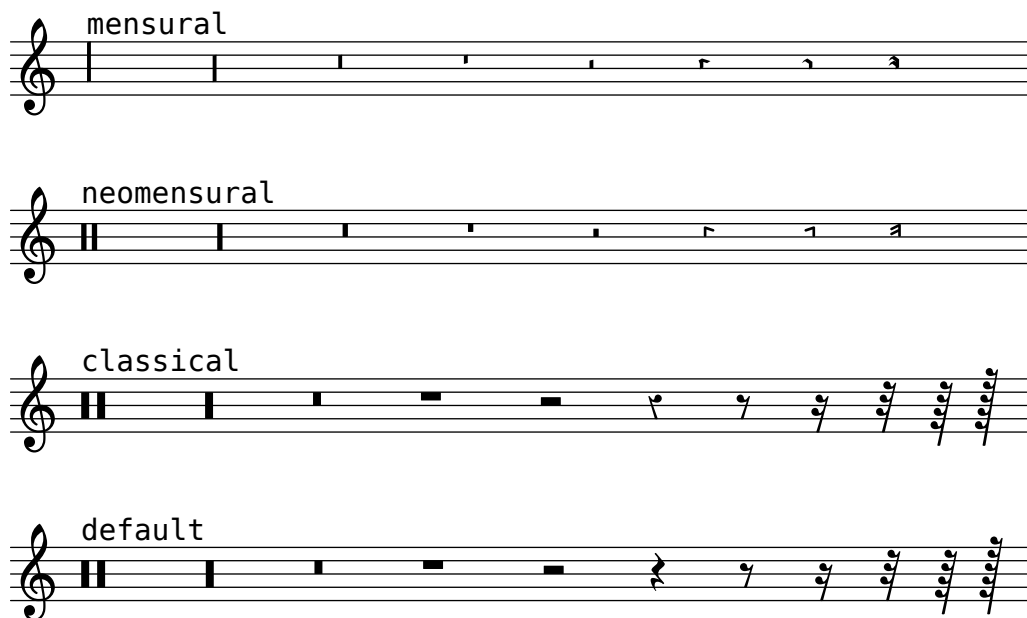
\new Staff \relative c {
  \cadenzaOn
  \override Staff.Rest #'style = #'mensural
  r\maxima^ \markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest #'style = #'neomensural
  r\maxima^ \markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest #'style = #'classical
  r\maxima^ \markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""

  \override Staff.Rest #'style = #'default
  r\maxima^ \markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
}

```



Voir aussi

Glossaire musicologique : Section “breve” dans *Glossaire*, Section “longa” dans *Glossaire*, Section “maxima” dans *Glossaire*.

Manuel de notation : [Silences valant une mesure], page 54.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Rest” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Il n'existe pas à proprement parler de limite (inférieure ou supérieure) en terme de durée d'un silence. Cependant, le nombre de glyphes disponibles ne couvre que les silences allant du trente-deuxième de soupir à la maxime (équivalant à huit pauses).

Silences invisibles

Un silence invisible – que l'on pourrait appeler un « saut » – peut être entré comme une note avec le nom de note **s** ou avec `\skip durée` :

```
c4 c s c
s2 c
```



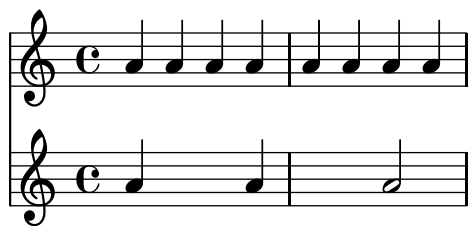
La syntaxe **s** est seulement disponible pour les modes d'entrée de notes et d'accords. Dans les autres situations, pour l'entrée de paroles par exemple, vous devrez utiliser la commande `\skip`, qui requiert une durée explicite ; cette durée ne sera pas prise en considération dès lors que les paroles suivent le rythme des notes de la mélodie à laquelle vous les aurez associées à l'aide des commandes `\addlyrics` ou `\lyricsto`.

```
<<
{
  a2 \skip2 a2 a2
}
\new Lyrics {
  \lyricmode {
    foo2 \skip 1 bla2
  }
}
>>
```



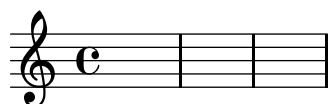
Gardez à l'esprit que `\skip` est une commande, et de ce fait n'affectera en rien la durée des notes qui suivent, contrairement à un **s**.

```
<<
{
  \repeat unfold 8 {a4}
}
{
  a4 \skip 2 a |
  s2 a
}
>>
```



La commande de saut génère simplement une case musicale vide. Le code de saut **s** crée tout de même les contextes **Staff** et **Voice** lorsque nécessaire, à l'instar des notes ou des silences :

```
s1 s s
```



Un `\skip` ne fait que sauter du temps musical ; il ne produit rien du tout, pas même un symbole transparent.

```
% This is valid input, but does nothing
\skip 1 \skip1 \skip 1
```

Voir aussi

Manuel d'initiation : [Section "Visibilité et couleur des objets"](#) dans *Manuel d'initiation*.

Manuel de notation : [\[Dictée à trous\]](#), page 204, [Section 5.4.6 \[Visibilité des objets\]](#), page 586.

Morceaux choisis : [Section "Rythme"](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section "SkipMusic"](#) dans *Référence des propriétés internes*.

Silences valant une mesure

Un silence valant une ou plusieurs mesures entières s'entre avec un **R** majuscule.

```
% Rest measures contracted to single measure
\compressFullBarRests
R1*4
R1*24
R1*4
b2^"Tutti" b4 a4
```



Ceci ne peut être utile que pour une mesure complètement vide. Sa durée doit donc correspondre à la longueur de la mesure telle que définie par la métrique. C'est la raison pour laquelle on utilisera aussi des points d'augmentation ou des fractions :

```
\compressFullBarRests
\time 2/4
R1 | R2 |
\time 3/4
R2. | R2.*2 |
\time 13/8
R1*13/8 | R1*13/8*12 |
\time 10/8
R4*5*4 |
```




Un R qui s'étend sur une seule mesure s'imprime tantôt comme une pause, tantôt comme une brève – ou « bâton de pause » – qui sera centrée sur la mesure quelle qu'en soit la métrique :

```
\time 4/4
R1 |
\time 6/4
R1*3/2 |
\time 8/4
R1*2 |
```



Par défaut, un silence multimesures sera répété sur autant de mesures que nécessaire. Il peut aussi n'être imprimé qu'une seule fois, surplombé du nombre de mesures vides – ou « à compter » :

```
% Default behavior
\time 3/4 r2. | R2.*2 |
\time 2/4 R2 |
\time 4/4
% Rest measures contracted to single measure
\compressFullBarRests
r1 | R1*17 | R1*4 |
% Rest measures expanded
\expandFullBarRests
\time 3/4
R2.*2 |
```



Vous pouvez aussi ajouter du texte à un silence multimesures en utilisant la syntaxe *note-markup* (cf. [Section 1.8.2 \[Mise en forme du texte\]](#), page 221). La variable `\fermataMarkup` quant à elle permet d'ajouter un point d'orgue :

```
\compressFullBarRests
\time 3/4
R2.*10^\markup { \italic "ad lib." }
R2.^\fermataMarkup
```



Note : C'est `MultiMeasureRestText` qui créera le texte, non `TextScript`. Les commandes de dérogation ou de redéfinition doivent s'adresser à l'objet concerné, comme vous pouvez le constater dans l'exemple suivant.

```
% Ceci échouera : il y a erreur quant à l'objet spécifié
\override TextScript #'padding = #5
R1~"pas bon !"
% Formulation correcte, qui fonctionnera
\override MultiMeasureRestText #'padding = #5
R1~"ça marche !"
```

ça marche !



Un silence multimesures placé directement après une commande `\partial` risque fort de perturber le vérificateur de limites et numéros de mesure.

Commandes prédéfinies

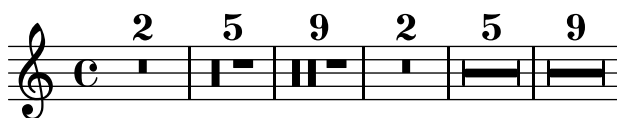
```
\textLengthOn,      \textLengthOff,      \fermataMarkup,      \compressFullBarRests,
\expandFullBarRests.
```

Morceaux choisis

Modifier l'apparence d'un silence multi-mesures

Dans le cas où ce silence dure moins de dix mesures, LilyPond imprime sur la portée des « silences d'église » – *Kirchenpause* en allemand – et qui sont une simple suite de rectangles. La propriété `expand-limit` permet d'obtenir un silence unique :

```
\relative c'' {
  \compressFullBarRests
  R1*2 | R1*5 | R1*9
  \override MultiMeasureRest #'expand-limit = #3
  R1*2 | R1*5 | R1*9
}
```



Positionnement des silences multimesures

Si l'on peut positionner verticalement un silence simple en le rattachant à une note, il n'en va pas de même pour un silence multimesure. Néanmoins, et uniquement dans le cadre de musique polyphonique, les silences multimesures sont positionnés différemment selon qu'ils appartiennent à une voix au numéro pair ou impair. Le positionnement des silences multimesures peut se contrôler ainsi :

```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line
  R1
  % They can be moved using an override
  \override MultiMeasureRest #'staff-position = #-2
  R1
  \override MultiMeasureRest #'staff-position = #0
  R1
  \override MultiMeasureRest #'staff-position = #2
```

```

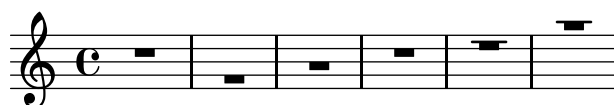
R1
\override MultiMeasureRest #'staff-position = #3
R1
\override MultiMeasureRest #'staff-position = #6
R1
\revert MultiMeasureRest #'staff-position
\break

% In two Voices, odd-numbered voices are under the top line
<< { R1 } \\\ { a1 } >>
% Even-numbered voices are under the bottom line
<< { a1 } \\\ { R1 } >>
% Multi-measure rests in both voices remain separate
<< { R1 } \\\ { R1 } >>

% Separating multi-measure rests in more than two voices
% requires an override
<< { R1 } \\\ { R1 } \\\
  \once \override MultiMeasureRest #'staff-position = #0
  { R1 }
>>

% Using compressed bars in multiple voices requires another override
% in all voices to avoid multiple instances being printed
\compressFullBarRests
<<
  \revert MultiMeasureRest #'staff-position
  { R1*3 }
  \\\
  \revert MultiMeasureRest #'staff-position
  { R1*3 }
>>
}

```



Ajout de texte à un silence multimesure

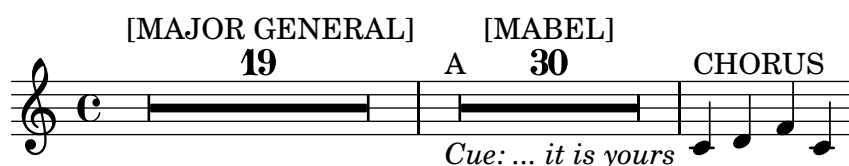
Lorsque du texte est attaché à un silence multimesure, il sera centré dans la mesure, au-dessus ou en dessous de la portée. Afin d'étirer la mesure dans le cas où ce texte est relativement long, il suffit d'insérer un silence invisible auquel on attache le texte en question, avant le silence multimesure.

Rappelez-vous qu'un silence invisible génère une barre de mesure. Le texte attaché à ce silence invisible sera alors aligné sur la gauche de là où serait positionnée la note. Cependant, si la taille de la mesure est déterminée par la longueur du texte, il apparaîtra comme centré.

```

\relative c' {
  \compressFullBarRests
  \textLengthOn
  s1*0^\markup { [MAJOR GENERAL] }
  R1*19
  s1*0_\markup { \italic { Cue: ... it is yours } }
  s1*0^\markup { A }
  R1*30^\markup { [MABEL] }
  \textLengthOff
  c4^\markup { CHORUS } d f c
}

```



Voir aussi

Glossaire musicologique : [Section “silence multimesures” dans *Glossaire*](#).

Manuel de notation : [\[Durées\]](#), page 40, [Section 1.8 \[Texte\]](#), page 212, [Section 1.8.2 \[Mise en forme du texte\]](#), page 221, [\[Commentaires textuels\]](#), page 213.

Morceaux choisis : [Section “Rythme” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “MultiMeasureRest” dans *Référence des propriétés internes*](#). [Section “MultiMeasureRestNumber” dans *Référence des propriétés internes*](#), [Section “MultiMeasureRestText” dans *Référence des propriétés internes*](#).

Problèmes connus et avertissements

Vous ne pouvez pas utiliser de doigtés (par ex. `R1*10-4`) pour positionner des nombres au dessus d’un silence multimesure, le numéro de doigt (4) risquant de chevaucher le nombre de mesures à compter (10).

Condenser plusieurs silences en un unique silence multimesures ne peut être automatisé.

Les silences multimesures peuvent générer des collisions avec d’autres silences.

1.2.3 Gravure du rythme

Métrique

Le chiffre de mesure indique le mètre d’une pièce : une alternance régulière de temps forts et de temps faibles. Il est indiqué par une fraction au début de la portée :

```

\time 2/4 c2
\time 3/4 c2.

```

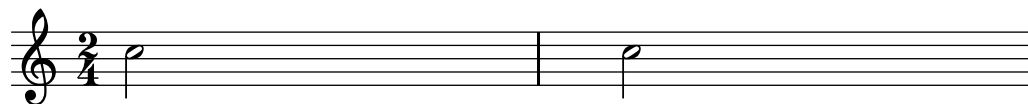


La métrique est imprimée en début de morceau, et à chaque fois qu’elle est modifiée. Si cette modification intervient au niveau d’un saut de ligne, une métrique « de précaution » sera imprimée avant de passer à la ligne suivante. Ce comportement par défaut peut être modifié, comme indiqué au chapitre [Section 5.4.6 \[Visibilité des objets\]](#), page 586.

```

\time 2/4
c2 c
\break
c c
\break
\time 4/4
c c c c

```



Le symbole de métrique utilisé pour les mesures à 2/2 et 4/4 peut être changé pour un style numérique :

```

% Default style
\time 4/4 c1
\time 2/2 c1
% Change to numeric style
\numericTimeSignature
\time 4/4 c1
\time 2/2 c1
% Revert to default style
\defaultTimeSignature
\time 4/4 c1
\time 2/2 c1

```



Les métriques anciennes font l'objet d'un [Section "chapitre particulier"](#) dans *Manuel de notation*.

En plus de déterminer la métrique qui sera imprimée, la commande `\time` réglera aussi les valeurs par défaut des propriétés `baseMoment`, `beatStructure` et `beamExceptions` correspondant à la métrique. Les valeurs prédéterminées par défaut de ces différentes propriétés sont inscrites dans le fichier '`scm/time-signature-settings.scm`'.

La valeur par défaut de `beatStructure` peut se voir aménagée dès la commande `\time` à l'aide d'un premier argument :

```

\score {
  \new Staff {
    \relative c' {
      \time #'(2 2 3) 7/8
      \repeat unfold 7 { c8 } |
    }
  }
}

```

```

\time #'(3 2 2) 7/8
\repeat unfold 7 { c8 } |
}
}
}

```



Les valeurs par défaut de toutes ces variables associées à la métrique, y compris `baseMoment` et `beamExceptions`, peuvent se définir en même temps. Ces valeurs peuvent se régler indépendamment pour différentes métriques. Les valeurs adaptées ne seront effectives qu'à partir du moment où interviendra une commande `\time` de la valeur de métrique correspondante :

```

\score {
  \relative c' {
    \overrideTimeSignatureSettings
      4/4          % timeSignatureFraction
      1/4          % baseMomentFraction
      #'(3 1)      % beatStructure
      #'()         % beamExceptions
    \time 4/4
    \repeat unfold 8 { c8 } |
  }
}

```



`\overrideTimeSignatureSettings` prend quatre arguments :

1. *timeSignatureFraction*, une fraction indiquant la métrique pour laquelle ces valeurs doivent s'appliquer,
2. *baseMomentFraction*, une fraction comprenant les numérateur et dénominateur de la fraction indiquant la base de la pulsation,
3. *beatStructure*, une liste Scheme indiquant la structure de cette pulsation, en unité de base,
4. *beamExceptions*, une liste associative des règles de ligature pour cette métrique, en dehors de celles basées sur le temps comme indiqué à la rubrique [Définition des règles de ligature automatique], page 78.

Le contexte auquel doivent s'appliquer les dérogations introduites par un `\overrideTimeSignatureSettings` doit exister avant l'appel de ces réglages. Autrement dit, vous devrez explicitement mentionner le contexte avant que ne survienne l'appel à `\overrideTimeSignatureSettings` s'il n'y a pas déjà d'élément musical.

```

\score {
  \relative c' {
    % This call will fail because the context isn't yet instantiated
    \overrideTimeSignatureSettings
      4/4          % timeSignatureFraction
      1/4          % baseMomentFraction

```

```

    #'(3 1)      % beatStructure
    #'()        % beamExceptions
\time 4/4
c8^\markup {"Beamed (2 2)"}
\repeat unfold 7 { c8 } |
% This call will succeed
\overrideTimeSignatureSettings
  4/4          % timeSignatureFraction
  1/4          % baseMomentFraction
  #'(3 1)      % beatStructure
  #'()        % beamExceptions
\time 4/4
c8^\markup {"Beamed (3 1)"}
\repeat unfold 7 { c8 } |
}
}

```



Vous pouvez revenir à tout moment aux réglages prédéterminés d'une métrique :

```

\score{
  \relative c' {
    \repeat unfold 8 { c8 } |
    \overrideTimeSignatureSettings
      4/4          % timeSignatureFraction
      1/4          % baseMomentFraction
      #'(3 1)      % beatStructure
      #'()        % beamExceptions
    \time 4/4
    \repeat unfold 8 { c8 } |
    \revertTimeSignatureSettings 4/4
    \time 4/4
    \repeat unfold 8 { c8 } |
  }
}

```



Le fait de déplacer du contexte `Score` au contexte `Staff` à la fois le `Timing_translator` et le `Default_bar_line_engraver` permet d'obtenir des réglages particuliers pour les différentes portées d'un regroupement :

```

\score {
  \new StaffGroup <<
    \new Staff {
      \overrideTimeSignatureSettings
        4/4          % timeSignatureFraction
        1/4          % baseMomentFraction
        #'(3 1)      % beatStructure
    }
  }
}

```

```

        #'()          % beamExceptions
    \time 4/4
    \repeat unfold 8 {c''8}
}
\new Staff {
    \overrideTimeSignatureSettings
        4/4          % timeSignatureFraction
        1/4          % baseMomentFraction
        #'(1 3)      % beatStructure
        #'()          % beamExceptions
    \time 4/4
    \repeat unfold 8 {c''8}
}
>>
\layout {
    \context {
        \Score
        \remove "Timing_translator"
        \remove "Default_bar_line_engraver"
    }
    \context {
        \Staff
        \consists "Timing_translator"
        \consists "Default_bar_line_engraver"
    }
}
}

```



Une autre méthode de modification de ces variables liées à la métrique, et qui évite sa réimpression au moment du changement, est indiquée à la rubrique [\[Définition des règles de ligature automatique\]](#), page 78.

Commandes prédéfinies

`\numericTimeSignature`, `\defaultTimeSignature`.

Morceaux choisis

Affichage seulement du numérateur d'une métrique (au lieu d'une fraction)

La métrique est parfois indiquée non pas par une fraction (par ex. 7/4) mais simplement par son numérateur (7 dans ce cas). L'instruction `\override Staff.TimeSignature #'style = #'single-digit` permet de déroger au style par défaut de manière permanente – un `\revert Staff.TimeSignature #'style` annulera ces modifications. Lorsque cette métrique sous la forme d'un seul chiffre ne se présente qu'une seule fois, il suffit de faire précéder l'instruction `\override` d'un simple `\once`.

```
\relative c'' {
```



```

\time 3/4
c4 c c
% Change the style permanently
\override Staff.TimeSignature #'style = #'single-digit
\time 2/4
c4 c
\time 3/4
c4 c c
% Revert to default style:
\revert Staff.TimeSignature #'style
\time 2/4
c4 c
% single-digit style only for the next time signature
\once \override Staff.TimeSignature #'style = #'single-digit
\time 5/4
c4 c c c c
\time 2/4
c4 c
}

```



Voir aussi

Glossaire musicologique : [Section “métrique”](#) dans *Glossaire*.

Manuel de notation : [\[Définition des règles de ligature automatique\]](#), page 78, [\[Métriques anciennes\]](#), page 413, [\[Gestion du temps\]](#), page 108.

Installed Files: ‘[scm/time-signature-settings.scm](#)’.

Morceaux choisis : [Section “Rythme”](#) dans *Morceaux choisis*.

Références des propriétés internes : [Section “TimeSignature”](#) dans *Référence des propriétés internes*, [Section “Timing_translator”](#) dans *Référence des propriétés internes*.

Indication métronomique

Une indication métronomique s’insère tout simplement comme ceci :

```

\tempo 4 = 120
c2 d
e4. d8 c2

```



Lorsque le réglage précis du métronome est laissé à l’appréciation de l’exécutant, vous pouvez cependant lui fournir une plage :

```

\tempo 4 = 40 ~ 46
c4. e8 a4 g
b,2 d4 r

```



Vous pouvez préférer une indication textuelle :

```
\tempo "Allegretto"
c4 e d c
b4. a16 b c4 r4
```



Lorsque vous combinez des indications métronomiques sous forme textuelle et numérique, l'indication numérique sera placée entre parenthèses :

```
\tempo "Allegro" 4 = 160
g4 c d e
d4 b g2
```



En matière d'indication textuelle, vous pouvez utiliser n'importe quel objet de type *markup*, comme ici :

```
\tempo \markup { \italic Faster } 4 = 132
a8-. r8 b-. r gis-. r a-. r
```



Mentionner une indication textuelle vide vous permet de mettre entre parenthèses l'indication numérique :

```
\tempo "" 8 = 96
d4 g e c
```



Morceaux choisis

Impression du métronome et des repères sous la portée

Les indications de tempo et les marques de repère s'impriment par défaut au-dessus de la portée. Le fait de régler en conséquence la propriété `direction` des objets `MetronomeMark` ou `RehearsalMark` les placera au-dessous de la portée.

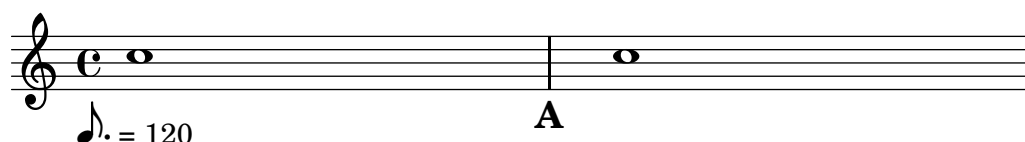
```

\layout { ragged-right = ##f }

{
  % Metronome marks below the staff
  \override Score.MetronomeMark #'direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark #'direction = #DOWN
  \mark \default
  c''1
}

```



Changement de tempo sans indication sur la partition

Vous pouvez indiquer un changement de tempo pour le fichier MIDI sans pour autant l'imprimer. Il suffit alors de le rendre invisible aux interprètes.

```

\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}

```



Création d'une indication métronomique sous forme d'étiquette

Vous pouvez créer des indications de tempo sous la forme d'étiquettes textuelles – des objets *markup* – notamment des équivalences. Cependant, elles n'apparaîtront pas dans le fichier MIDI.

```

\relative c' {
  \tempo \markup {
    \concat {
      (
        \smaller \general-align #Y #DOWN \note #"16." #1
        " = "
        \smaller \general-align #Y #DOWN \note #"8" #1
      )
    }
  }
}

```

```

      }
    }
  c1
  c4 c' c,2
}

```



Pour de plus amples détails, veuillez consulter [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

Voir aussi

Glossaire musicologique : [Section “métronome”](#) dans *Glossaire*, [Section “indication métronomique”](#) dans *Glossaire*, [Section “indication du tempo”](#) dans *Glossaire*, [Section “marque de métronome”](#) dans *Glossaire*.

Manuel de notation : [Section 1.8.2 \[Mise en forme du texte\]](#), page 221, [Section 3.5 \[Sortie MIDI\]](#), page 485.

Morceaux choisis : [Section “Notation sur la portée”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “MetronomeMark”](#) dans *Référence des propriétés internes*.

Levées

Les mesures incomplètes, telles que les anacrouses ou levées, doivent être entrées avec la commande `\partial`. La syntaxe de cette commande est

```
\partial durée
```

où *durée* correspond à la valeur rythmique **restant** dans la mesure et **avant** la première mesure complète :

```

\time 3/4
\partial 8
e8 | a4 c8 b c4 |

```



La *durée* peut être de n'importe quelle longueur inférieure à la mesure complète :

```

\time 3/4
\partial 4.
r4 e8 | a4 c8 b c4 |

```



Cette commande `\partial durée` peut aussi s'écrire :

```
\set Timing.measurePosition = -durée
```

De fait, `\partial 8*3` équivaut à :

```
\time 3/4
\set Timing.measurePosition = #(ly:make-moment -1 8)
e8 | a4 c8 b c4 |
```



La propriété `measurePosition` contient un nombre rationnel qui indique, à ce point précis, où l'on en est de la mesure. Notez qu'il s'agit du résultat d'une soustraction ; `\partial 4` signifie, pour le programme : « Dans cette mesure, il reste juste une noire ».

Voir aussi

Glossaire musicologique : [Section “anacrouse” dans *Glossaire*](#).

Manuel de notation : [\[Notes d'ornement\]](#), page 102.

Morceaux choisis : [Section “Rythme” dans *Morceaux choisis*](#).

Références des propriétés internes : [Section “Timing_translator” dans *Référence des propriétés internes*](#).

Problèmes connus et avertissements

`\partial` n'est destiné à être utilisé qu'en début de pièce. Si on l'utilise ailleurs qu'au début, des messages d'erreur peuvent s'afficher, voire des problèmes inopinés surgir. Utilisez plutôt `\set Timing.measurePosition` en pareil cas.

```
\time 6/8
\partial 8
e8 | a4 c8 b[ c b] |
\set Timing.measurePosition = #(ly:make-moment -1 4)
r8 e,8 | a4 c8 b[ c b] |
```



Musique sans métrique

En matière de musique mesurée, le positionnement des barres et la numérotation sont calculés automatiquement. Ceci n'est pas souhaitable dans le cas d'une musique non mesurée – les cadences par exemple – et peut être désactivé à l'aide de la commande `\cadenzaOn`, puis réactivé par un `\cadenzaOff` à l'endroit approprié.

```
c4 d e d
\cadenzaOn
c4 c d8[ d d] f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```



La numérotation reprend dès la fin d'une cadence.

```
% Show all bar numbers
\override Score.BarNumber #'break-visibility = #all-visible
c4 d e d
\cadenzaOn
c4 c d8[ d d] f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```



Une cadence représente un fragment détaché de la notion de mesure, en dépit des différentes commandes `\bar` qu'il pourra comporter. Ceci a pour conséquence que toute altération supplémentaire par rapport à l'armure devra être rappelée manuellement – voir [\[Altérations\]](#), page 5.

```
c4 d e d
\cadenzaOn
cis4 d cis d
\bar "|"
cis4 d cis! d
\cadenzaOff
\bar "|"
```



L'instruction `\cadenzaOn` désactive les ligatures automatiques ; elles seront réactivées après un `\cadenzaOff`. Toutes les ligatures devront donc être indiquées de manière explicite tout au long de la cadence – voir [\[Barres de ligature manuelles\]](#), page 86).

```
\repeat unfold 8 { c8 }
\cadenzaOn
cis8 c c c c
\bar"|"
c8 c c
\cadenzaOff
\repeat unfold 8 { c8 }
```



Ces commandes prédéfinies affecteront toutes les portées de la partition, même si vous ne les placez que dans un simple contexte `Voice`. Pour éviter ce désagrément, transférez le `Timing_translator` du contexte `Score` au contexte `Staff`, comme indiqué au chapitre [\[Notation polymétrique\]](#), page 69.

Commandes prédéfinies

`\cadenzaOn`, `\cadenzaOff`.

Voir aussi

Glossaire musicologique : [Section “cadence”](#) dans *Glossaire*.

Manuel de notation : [Section 5.4.6 \[Visibilité des objets\]](#), page 586, [\[Notation polymétrique\]](#), page 69, [\[Barres de ligature manuelles\]](#), page 86, [\[Altérations\]](#), page 5.

Morceaux choisis : [Section “Rythme”](#) dans *Morceaux choisis*.

Problèmes connus et avertissements

Sauts de ligne ou de page ne peuvent intervenir qu’au niveau d’une barre de mesure. Si votre musique non mesurée s’étend sur plus d’une ligne, il vous faudra insérer des barres de mesure « invisibles » pour indiquer où des sauts de ligne peuvent prendre place :

```
\bar ""
```

Au risque d’obtenir des messages d’erreur inhabituels, il est fortement recommandé de créer explicitement un contexte `Voice` lorsqu’un morceau débute par un `\cadenzaOn`.

```
\new Voice {
  \relative c' {
    \cadenzaOn
    c16[^"Solo Free Time" d e f] g2.
    \bar "||"
    \cadenzaOff
  }
}
```

Notation polymétrique

LilyPond gère les métriques composites, aussi bien de manière explicite que de manière détournée – modification de l’indicateur de métrique et échelonnement de la durée des notes.

Métriques différentes et mesures d’égale longueur

Il suffit, pour obtenir cette forme de notation, de tout d’abord affecter une même métrique aux différentes portées. Cette métrique sera ensuite remplacée dans chacune des portées par un quotient fourni en argument à la propriété `timeSignatureFraction`. La durée des notes sera enfin proratisée selon la métrique commune grâce à la fonction `\scaleDurations`.

L’exemple suivant utilise parallèlement des mesures à 3/4, 9/8 et 10/8. Pour la deuxième portée les durées sont multipliées par 2/3 de telle sorte que $2/3 * 9/8 = 3/4$; pour la troisième elles sont multipliées par 3/5, de telle sorte que $3/5 * 10/8 = 3/4$. Les ligatures devront être explicites, la fonction d’échelonnement venant perturber les règles de ligature automatique.

```
\relative c' <<
  \new Staff {
    \time 3/4
    c4 c c |
    c4 c c |
  }
  \new Staff {
    \time 3/4
    \set Staff.timeSignatureFraction = 9/8
    \scaleDurations 2/3
    \repeat unfold 6 { c8[ c c] }
  }
  \new Staff {
    \time 3/4
```

```

\set Staff.timeSignatureFraction = 10/8
\scaleDurations 3/5 {
  \repeat unfold 2 { c8[ c c] }
  \repeat unfold 2 { c8[ c] } |
  c4. c \times 2/3 { c8[ c c] } c4
}
}
>>

```



Métriques différentes et mesures de longueur inégale

Il arrive aussi que chaque portée ait sa propre métrique. Vous y parviendrez en déplaçant le `Timing_translator` et le `Default_bar_line_engraver` dans le contexte `Staff`.

```

\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
}

% Now each staff has its own time signature.

\relative c' <<
  \new Staff {
    \time 3/4
    c4 c c |
    c4 c c |
  }
  \new Staff {
    \time 2/4
    c4 c |
    c4 c |
    c4 c |
  }
  \new Staff {
    \time 3/8

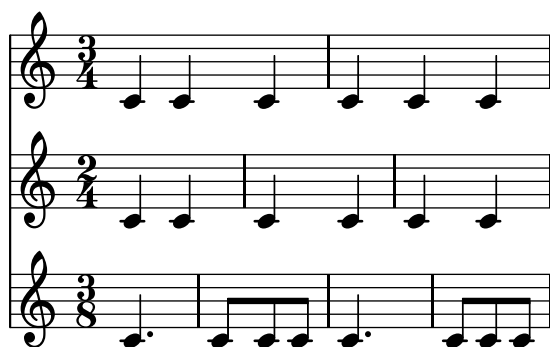
```



```

c4. |
c8 c c |
c4. |
c8 c c |
}
>>

```



Métriques complexes

Une métrique composite se crée à l'aide de la fonction `\compoundMeter`, en suivant la syntaxe :

```
\compoundMeter #'(liste de listes)
```

La construction la plus simple est constituée d'une seule liste, dans laquelle le *dernier* nombre indique le « dénominateur » de la métrique, les précédents représentent le « numérateur ».

```

\relative c' {
  \compoundMeter #'((2 2 2 8))
  \repeat unfold 6 c8 \repeat unfold 12 c16
}

```



Une métrique plus élaborée s'obtient en ajoutant d'autres listes. Bien entendu, les ligatures automatiques s'ajusteront aux différentes valeurs.

```

\relative c' {
  \compoundMeter #'((1 4) (3 8))
  \repeat unfold 5 c8 \repeat unfold 10 c16
}

```

```

\relative c' {
  \compoundMeter #'((1 2 3 8) (3 4))
  \repeat unfold 12 c8
}

```



Voir aussi

Glossaire musicologique : Section “polymétrie” dans *Glossaire*, Section “métrique composite” dans *Glossaire*, Section “métrique” dans *Glossaire*.

Manuel de notation : [Barres de ligature automatiques], page 75, [Barres de ligature manuelles], page 86, [Métrique], page 58, [Changement d’échelle des durées], page 46.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Références des propriétés internes : Section “TimeSignature” dans *Référence des propriétés internes*, Section “Timing_translator” dans *Référence des propriétés internes*, Section “Default_bar_line_engraver” dans *Référence des propriétés internes*, Section “Staff” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

L’utilisation de métriques différentes en parallèle entraîne un alignement vertical. De ce fait, les barres de mesure ont tendance à fausser l’espacement régulier entre les différentes portées.

Découpage automatique des notes

On peut convertir automatiquement les notes longues en notes liées. Il suffit pour cela de remplacer le graveur `Note_heads_engraver` par le graveur `Completion_heads_engraver`. Il en va de la même manière pour des silences ; le `Completion_rest_engraver` devra alors remplacer le `Rest_engraver`. Dans l’exemple suivant, les notes ou silences qui dépassent de la mesure sont divisés et au besoin liés.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
  \remove "Rest_engraver"
  \consists "Completion_rest_engraver"
}

{ c2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2 r1*2 }
```



Ces graveurs divisent toutes les notes et silences qui débordent de la mesure, et insèrent des liaisons de prolongation. Dans la pratique, cette fonctionnalité permet de déboguer des partitions complexes : si les mesures ne sont pas entièrement remplies, alors les liaisons de prolongation montrent exactement la durée des décalages de mesure.

Voir aussi

Glossaire musicologique : Section “liaison de tenue” dans *Glossaire*.

Manuel d’initiation : Section “Tout savoir sur les graveurs” dans *Manuel d’initiation*, Section “Ajout et suppression de graveurs” dans *Manuel d’initiation*.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Références des propriétés internes : Section “Note_heads_engraver” dans *Référence des propriétés internes*, Section “Completion_heads_engraver” dans *Référence des propriétés internes*.

propriétés internes, Section “Rest_engraver” dans *Référence des propriétés internes*, Section “Completion_rest_engraver” dans *Référence des propriétés internes*, Section “Forbid_line_break_engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

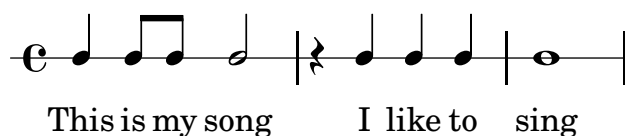
Bien que toutes les durées – particulièrement celles contenant des nolets – ne puissent être exactement représentées avec des notes normales et des points, le graveur `Completion_heads_engraver` n’insérera pas de nolet.

Le `Completion_heads_engraver` affecte seulement les notes, il ne divise pas les silences.

Gravure de lignes rythmiques

Au moyen d’une portée rythmique – *rhythmic staff* en anglais – on peut montrer seulement le rythme d’une mélodie : toutes les notes sont ramenées à la même hauteur, sur une portée d’une seule ligne.

```
<<
\new RhythmicStaff {
  \new Voice = "myRhythm" {
    \time 4/4
    c4 e8 f g2
    r4 g g f
    g1
  }
}
\new Lyrics {
  \lyricsto "myRhythm" {
    This is my song
    I like to sing
  }
}
>>
```



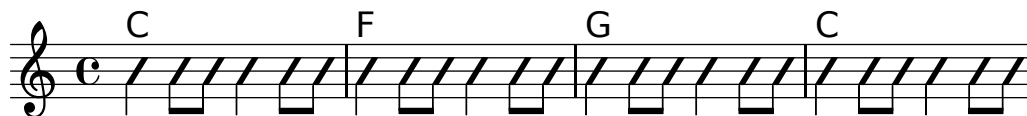
L’utilisation combinée du `Pitch_squash_engraver` et de `\improvisationOn` permet d’afficher la structure rythmique d’une grille d’accords :

```
<<
\new ChordNames {
  \chordmode {
    c1 f g c
  }
}
\new Voice \with {
  \consists "Pitch_squash_engraver"
} \relative c'' {
  \improvisationOn
  c4 c8 c c4 c8 c
  f4 f8 f f4 f8 f
  g4 g8 g g4 g8 g
}
```

```

    c4 c8 c c4 c8 c
  }
>>

```



Commandes prédéfinies

`\improvisationOn`, `\improvisationOff`.

Morceaux choisis

Rythmique et guitare

En matière de notation pour guitare, il arrive que soient indiqués les « coups de gratte » en plus de la mélodie, grilles d'accords et diagrammes de tablature.

```
\include "predefined-guitar-fretboards.ly"
```

```

<<
  \new ChordNames {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new FretBoards {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } {
    \relative c'' {
      \improvisationOn
      c4 c8 c c4 c8 c
      f4 f8 f f4 f8 f
      g4 g8 g g4 g8 g
      c4 c8 c c4 c8 c
    }
  }
  \new Voice = "melody" {
    \relative c'' {
      c2 e4 e4
      f2. r4
      g2. a4
      e4 c2.
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      This is my song.
      I like to sing.
    }
  }

```

}
 }
 >>

C F G
 32 1 134211 21 3
 This is my song. I like
 to sing.

Voir aussi

Morceaux choisis : [Section “Rythme”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “RhythmicStaff”](#) dans *Référence des propriétés internes*. [Section “Pitch_squash_engraver”](#) dans *Référence des propriétés internes*.

1.2.4 Barres de ligature

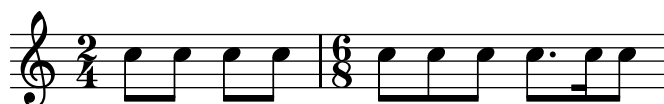
Barres de ligature automatiques

LilyPond décide automatiquement de la manière de grouper les notes et d’imprimer les ligatures.

```

\time 2/4 c8 c c c
\time 6/8 c8 c c c8. c16 c8

```



Lorsque ce comportement automatisé n’est pas satisfaisant, on peut définir des groupements manuellement – voir [\[Barres de ligature manuelles\]](#), page 86. Dans le cas où le groupe de notes en question contient un silence, il est **impératif** d’indiquer explicitement les début et fin de la ligature.

Lorsque les ligatures automatiques ne sont pas nécessaires, il suffit de désactiver la fonctionnalité par un `\autoBeamOff` – réactivation par `\autoBeamOn` :

```

c4 c8 c8. c16 c8. c16 c8
\autoBeamOff
c4 c8 c8. c16 c8.
\autoBeamOn
c16 c8

```



Note : Si des ligatures sont utilisées dans les paroles d’une chanson (pour indiquer des méliques), les ligatures automatiques doivent être désactivées, avec `\autoBeamOff`, et indiquées manuellement. L’utilisation conjointe de `\partcombine` et de `\autoBeamOff` peut produire des résultats quelque peu surprenants ; ceci fait l’objet d’un exemple particulier à la rubrique morceaux choisis.

Des règles de dérogation au comportement automatique par défaut sont possibles ; voir [\[Définition des règles de ligature automatique\]](#), page 78.

Commandes prédéfinies

`\autoBeamOff`, `\autoBeamOn`.

Morceaux choisis

Ligature au moment d’un saut de ligne

Il est normalement impensable qu’un saut de ligne tombe au milieu d’une ligature. LilyPond permet néanmoins de l’obtenir.

```

\relative c'' {
  \override Beam #'breakable = ##t
  c8 c[ c] c[ c] c[ c] c[ \break
  c8] c[ c] c[ c] c[ c] c
}

```



Ligature et directions de hampe inversées

LilyPond insère automatiquement des ligatures coudées – certaines hampes vers le haut, d’autres vers le bas – lorsqu’il détecte un intervalle important entre des têtes de notes. Ce comportement peut être changé par l’intermédiaire de l’objet `auto-knee-gap` – défini par défaut à ‘5,5’ espace, plus la largeur et la pente de la ligature en question.

```

{
  f8 f''8 f8 f''8
  \override Beam #'auto-knee-gap = #6
  f8 f''8 f8 f''8
}

```

}



Partcombine et autoBeamOff

La fonction `\autoBeamOff` dans le cadre d'un `\partcombine` agit de façon bien particulière ; c'est pourquoi il vaut mieux tout d'abord recourir à

```
\set Staff.autobeaming = ##f
```

pour désactiver les ligatures automatiques pour l'ensemble de la portée concernée.

L'instruction `\partcombine` fonctionne apparemment sur la base de trois voix : solo hampes montantes, solo hampes descendantes et ensemble hampes montantes.

Lorsque `\autoBeamOff` apparaît dans le premier argument de la combinaison, il s'applique à la voix active à ce moment précis, qu'il s'agisse du solo hampes montantes ou du combiné hampes montantes. Lorsqu'elle est introduite dans le second argument, la commande `\autoBeamOff` s'appliquera au solo hampes descendantes.

Vous devrez donc, afin que `\autoBeamOff` soit pleinement opérationnel dans le cadre d'un `\partcombine`, l'introduire aux **trois** niveaux.

```
{
  \% \set Staff.autoBeaming = ##f % turns off all autobeaming
  \partcombine
  {
    \autoBeamOff % applies to split up stems
    \repeat unfold 4 a'16
    \% \autoBeamOff % applies to combined up stems
    \repeat unfold 4 a'8
    \repeat unfold 4 a'16
  }
  {
    \autoBeamOff % applies to down stems
    \repeat unfold 4 f'8
    \repeat unfold 8 f'16 |
  }
}
```



Voir aussi

Manuel de notation : [Barres de ligature manuelles], page 86, [Définition des règles de ligature automatique], page 78.

Fichiers d'initialisation : 'scm/auto-beam.scm'.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Références des propriétés internes : Section "Auto_beam_engraver" dans *Référence des propriétés internes*, Section "Beam_engraver" dans *Référence des propriétés internes*, Section "Beam" dans *Référence des propriétés internes*. Section "BeamEvent" dans

Référence des propriétés internes, Section “BeamForbidEvent” dans *Référence des propriétés internes*, Section “beam-interface” dans *Référence des propriétés internes*, Section “unbreakable-spanner-interface” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les propriétés d’une ligature sont déterminées **dès le début** de sa construction ; toute adaptation qui interviendrait avant sa terminaison ne sera prise en compte qu’à l’occasion de la **prochaine** ligature.

Définition des règles de ligature automatique

Lorsque la fonction de ligature automatique est active, le positionnement des ligatures dépend des trois propriétés `baseMoment`, `beatStructure` et `beamExceptions`. Les valeurs par défaut de ces variables peuvent s’adapter, comme indiqué ci-après, ou bien carrément être modifiées – voir [Métrique], page 58.

Dès lors qu’une règle affectant `beamExceptions` est définie pour la métrique en vigueur, c’est cette règle qui servira à déterminer le placement des ligatures. En l’absence de règle affectant `beamExceptions` pour la métrique en vigueur, les ligatures seront déterminées par les réglages conjoints de `baseMoment` et `beatStructure`.

`beamExceptions` dispose par défaut de règles pour les métriques le plus courantes ; il est donc impératif de les invalider pour gérer les ligatures automatiques à l’aide de `baseMoment` et `beatStructure`. Les règles de `beamExceptions` se désactivent par un

```
\set Timing.beamExceptions = #'()
```

Ligature basée sur `baseMoment` et `beatStructure`

Dans la plupart des cas, une ligature automatique se termine à la fin d’une pulsation. Ce sont les propriétés de contexte `baseMoment` et `beatStructure` qui détermineront là où prend fin la pulsation. La propriété `beatStructure` est constituée d’une liste d’éléments Scheme qui définit la longueur de chaque pulsation, prenant `baseMoment` comme unité. L’unité de base (le `baseMoment`) correspond par défaut à l’inverse du dénominateur de la métrique. D’autre part, chaque unité de `baseMoment` constitue une seule pulsation.

```
\time 5/16
c16^"default" c c c c |
\set Timing.beatStructure = #'(2 3)
c16^"(2+3)" c c c c |
\set Timing.beatStructure = #'(3 2)
c16^"(3+2)" c c c c |
```



Dans le cas d’une mesure à 4/4, `beamExceptions` **doit** être invalidé pour permettre à `beatStructure` de produire ses effets. La commande `\set Timing.beamExceptions = #'()` peut intervenir à tout moment, avant de déterminer les ligatures par `beatStructure`.

```
\time 4/4
a8^"default" a a a a a a

\set Timing.baseMoment = #(ly:make-moment 1 4)
\set Timing.beatStructure = #'(1 1 1 1)
a8^"no change" a a a a a a
```



```

\set Timing.beamExceptions = #'()
\set Timing.baseMoment = #(ly:make-moment 1 4)
\set Timing.beatStructure = #'(1 1 1 1)
a8^"changed" a a a a a a a

```



Les effets de règles de ligature peuvent être restreints à un contexte particulier. En l'absence de règle particulière déterminée dans un contexte de niveau inférieur, les règles définies au niveau directement supérieur s'appliqueront.

```

\new Staff {
  \time 7/8
  \set Staff.beatStructure = #'(2 3 2)
  <<
    \new Voice = one {
      \relative c'' {
        a8 a a a a a a
      }
    }
    \new Voice = two {
      \relative c' {
        \voiceTwo
        \set Voice.beatStructure = #'(1 3 3)
        f8 f f f f f f
      }
    }
  >>
}

```



Lorsque plusieurs voix cohabitent sur une même portée et que les règles de ligature doivent s'appliquer sans distinction, il faut spécifier que ces règles affectent le contexte **Staff** :

```

\time 7/8
% rhythm 3-1-1-2
% Context Voice specified -- does not work correctly
% Because of autogenerated voices, all beaming will
% be at baseMoment (1 . 8)
\set beatStructure = #'(3 1 1 2)
<< {a8 a a a16 a a a a8 a} \\ {f4. f8 f f f} >>

% Works correctly with context Staff specified
\set Staff.beatStructure = #'(3 1 1 2)
<< {a8 a a a16 a a a a8 a} \\ {f4. f8 f f f} >>

```



Vous pouvez ajuster la valeur de `baseMoment` afin d'obtenir des ligatures selon vos besoin. Notez cependant que la valeur de `beatStructure` devra être en corrélation avec cette nouvelle valeur de `baseMoment`.

```
\time 5/8
\set Timing.baseMoment = #(ly:make-moment 1 16)
\set Timing.beatStructure = #'(7 3)
\repeat unfold 10 { a16 }
```



`baseMoment` constitue un **moment**, autrement dit une unité de durée musicale. La fonction Scheme `ly:make-moment` est tout particulièrement chargée de créer cette quantité de type *moment* – voir [Gestion du temps], page 108 pour plus de précisions.

La pulsation – *baseMoment* en anglais – découle directement de la métrique telle que définie par la commande `\time`. Elle est par défaut égale à un sur le dénominateur de la métrique. Les exceptions à cette règle par défaut sont répertoriées dans le fichier ‘`scm/time-signature-settings.scm`’. Pour savoir comment jouer avec la valeur de `baseMoment` selon la métrique, reportez vous au chapitre [Métrique], page 58.

Les règles de ligature et de subdivision spécifiques sont enregistrées dans la propriété `beamExceptions`. Ses valeurs par défaut, rangées par métrique et type de règle, sont répertoriées dans le fichier ‘`scm/time-signature-settings.scm`’.

Ligature basée sur `beamExceptions`

Les règles spécifiques autres que celles concernant la terminaison des ligatures sont gérées par la propriété `beamExceptions`.

```
\time 3/16
\set Timing.beatStructure = #'(2 1)
\set Timing.beamExceptions =
  #'(
    (end . ;ouvre de la liste associative
      ( ;ceci concerne la terminaison des ligatures
        ((1 . 32) . (2 2 2)) ;ouvre la liste des points de terminaison
      )) ;règle pour les triples -- groupées à la double
    ) ;referme la liste
c16 c c |
\repeat unfold 6 { c32 } |
```



`beamExceptions` contient la liste des règles de ligature selon leur type.

Le seul type de règle pris en charge à ce jour est 'end, qui concerne les terminaisons.

Chaque règle est constituée d'une liste de doublets associatifs en langage Scheme (un *alist* pour les puristes), qui indique la durée de base et sa règle de regroupement.

```
#'((durée-type1 . groupement-1)
   (durée-type2 . groupement-2)
   (durée-type3 . groupement-3))
```

durée-type est constitué d'une paire indiquant la durée de base – par exemple (1 . 16) pour une double croche.

groupement est constitué d'une liste Scheme qui indique le regroupement à effectuer, en unité de type de ligature.

Note : La propriété `beamExceptions` doit contenir absolument **toutes** les exceptions. Il n'est en effet pas possible d'en ajouter, modifier ou supprimer *a posteriori*. Cela peut paraître fastidieux, mais toutes les règles de ligature devraient être appréciées avant de les spécifier.

Lorsqu'intervient un changement de métrique, les valeurs par défaut de `Timing.baseMoment`, `Timing.beatStructure` et `Timing.beamExceptions` sont réinitialisées. Il suffit donc, pour revenir aux règles de ligature par défaut d'un contexte `Timing`, de spécifier à nouveau la métrique.

```
\time 6/8
\repeat unfold 6 { a8 }
% group (4 + 2)
\set Timing.beatStructure = #'(4 2)
\repeat unfold 6 { a8 }
% go back to default behavior
\time 6/8
\repeat unfold 6 { a8 }
```



Les règles de ligature automatique par défaut sont répertoriées, par métrique, dans le fichier '`scm/time-signature-settings.scm`'. Les manières de déroger à ce comportement sont abordées au chapitre [Métrique], page 58.

De nombreuses règles de ligature automatique comportent une clé `beamExceptions`. Par exemple, s'il n'y a que des croches dans une mesure à 4/4, celles-ci seront réparties en deux groupes. Le fait de ne pas réinitialiser `beamExceptions` lors d'un aménagement de la pulsation – l'élément `beatStructure` – empêchera l'application de cette dérogation.

```
\time 4/4
\set Timing.baseMoment = #(ly:make-moment 1 8)
\set Timing.beatStructure = #'(3 3 2)
% This won't beam (3 3 2) because of beamExceptions
\repeat unfold 8 {c8} |
% This will beam (3 3 2) because we clear beamExceptions
\set Timing.beamExceptions = #'()
\repeat unfold 8 {c8}
```



De la même manière, les croches d'une mesure à 3/4 sont ligaturées sur la mesure par défaut. Ligaturer sur le temps requiert un appel à `beamExceptions`.

```
\time 3/4
% by default we beam in (6) due to beamExceptions
\repeat unfold 6 {a8} |
```

```
% This will beam (1 1 1) due to beatLength
\set Timing.beamExceptions = #'()
\repeat unfold 6 {a8}
```



Certaines partitions des périodes romantique ou classique font apparaître des ligatures sur la moitié d'une mesure à 3/4 (ou à 3/8), ce qui va à l'encontre de la règle établie – comme le fait remarquer Gould à la page 153 de son ouvrage – puisque donne l'impression d'une mesure à 6/8. Il en va de même pour une mesure à 3/8. La reproduction d'un tel comportement se contrôle à l'aide de la propriété de contexte `beamHalfMeasure`, qui d'ailleurs ne sera effective que lorsque le numérateur de la métrique est un 3.

```
\time 3/4
r4. a8 a a |
\set Timing.beamHalfMeasure = ##f
r4. a8 a a |
```



Principes de la ligature automatique

Lorsqu'elle est activée, la gestion automatisée des ligatures est directement liée aux propriétés `baseMoment`, `beatStructure` et `beamExceptions`.

Les règles déterminant le positionnement des ligatures automatiques s'appliquent dans l'ordre suivant de priorité :

- Une ligature explicite – indiquée par [...] – sera toujours respectée ; sinon
- si une règle explicite de terminaison a été définie grâce à la propriété `beamExceptions` pour un type de ligature spécifique dans la métrique en cours, c'est elle qui s'appliquera ; sinon
- si une règle explicite de terminaison a été définie grâce à la propriété `beamExceptions` pour un type de ligature plus large, c'est elle qui s'appliquera ; sinon
- utilisation des valeurs de `baseMoment` et `beatStructure` pour regrouper les notes par des ligatures.

Le *type de ligature* correspond à la durée la plus courte dans le groupe.

Les règles de ligature par défaut sont répertoriées dans le fichier '`scm/time-signature-settings.scm`'.

Morceaux choisis

Subdivision des ligatures

Les ligatures d'une succession de notes de durée inférieure à la croche ne sont pas subdivisées par défaut. Autrement dit, tous les traits de ligature (deux ou plus) seront continus. Ce comportement peut être modifié afin de diviser la ligature en sous-groupes grâce à la propriété `subdivideBeams`. Lorsqu'elle est activée, les ligatures seront subdivisées selon un intervalle défini par `baseMoment` ; il n'y aura alors plus qu'un seul trait de ligature entre chaque sous-groupe. Par défaut, `baseMoment` fixe la valeur de référence par rapport à la métrique en vigueur. Il faudra donc lui fournir, à l'aide de la fonction `ly:make-moment`, une fraction correspondant à la durée du sous-groupe désiré comme dans l'exemple ci-dessous. Gardez à l'esprit que, si vous venez à modifier `baseMoment`, vous devrez probablement adapter `beatStructure` afin qu'il reste en adéquation avec les nouvelles valeurs de `baseMoment`.

```

\relative c'' {
  c32[ c c c c c c c]
  \set subdivideBeams = ##t
  c32[ c c c c c c c]

  % Set beam sub-group length to an eighth note
  \set baseMoment = #(ly:make-moment 1 8)
  \set beatStructure = #'(2 2 2 2)
  c32[ c c c c c c c]

  % Set beam sub-group length to a sixteenth note
  \set baseMoment = #(ly:make-moment 1 16)
  \set beatStructure = #'(4 4 4 4)
  c32[ c c c c c c c]
}

```



Ligature à la pulsation

Une sous-ligature tronquée peut pointer en direction de la pulsation à laquelle elle se rattache. Dans l'exemple suivant, la première ligature évite toute troncature (comportement par défaut), alors que la deuxième respecte rigoureusement la pulsation.

```

\relative c'' {
  \time 6/8
  a8. a16 a a
  \set strictBeatBeaming = ##t
  a8. a16 a a
}

```



Signes de direction, signes de sous-groupe

Les règles de ligature par mesure sont gérées par la propriété `beatStructure`. Ses valeurs par défaut sont répertoriées, par métrique, dans le fichier `scm/time-signature-settings.scm`. Elles sont modifiables grâce à la commande `\set`. La fonction Scheme `set-time-signature` permet quant à elle de définir à la fois la métrique et la pulsation. Celle-ci prend trois arguments : le nombre de pulsations, la durée de la pulsation et le regroupement des pulsations dans la mesure. `\time` et `set-time-signature` s'appliquent tous deux au contexte `Timing` ; ils ne redéfiniront donc pas les valeurs de `beatStructure` ou `baseMoment` lorsqu'elles sont modifiées dans un contexte de niveau inférieur comme `Voice` par exemple.

Si l'on fait appel au `Measure_grouping_engraver`, la fonction `set-time-signature` créera aussi des symboles `MeasureGrouping`. Ces symboles aident à la lecture des œuvres modernes à la rythmique complexe. Dans l'exemple qui suit, la mesure à 9/8 est divisée en 2, 2, 2 et 3, alors que la mesure à 5/8 répond aux règles par défaut contenues dans le fichier `scm/beam-settings.scm`.

```

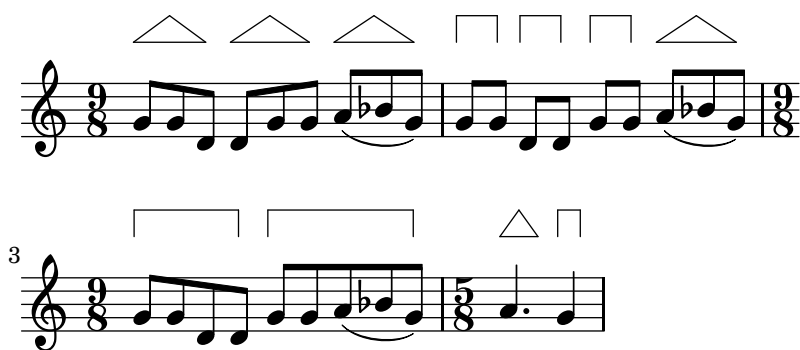
\score {
  \new Voice \relative c'' {
    \time 9/8

```

```

g8 g d d g g a( bes g) |
\set Timing.beatStructure = #'(2 2 2 3)
g8 g d d g g a( bes g) |
\time #'(4 5) 9/8
g8 g d d g g a( bes g) |
\time 5/8
a4. g4 |
}
\layout {
  \context {
    \Staff
    \consists "Measure_grouping_engraver"
  }
}
}

```



Définition de règles de ligature pour la partition

Les règles de ligature définies au niveau du contexte **Score** s'appliqueront à toutes les portées. Il est toutefois possible de moduler au niveau **Staff** ou **Voice** :

```

\relative c'' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.baseMoment = #(ly:make-moment 1 8)
  \set Score.beatStructure = #'(3 4 3)
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modify beaming for just this staff
      \set Staff.beatStructure = #'(6 4)
      c8 c c c c c c c c c
    }
    \new Staff {
      % Inherit beaming from Score context
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
      % Modify beaming for this voice only
    }
  >>
}

```

```

\new Voice {
  \voiceTwo
  \set Voice.beatStructure = #'(6 4)
  a8 a a a a a a a a
}
>>
}
>>
}

```



Voir aussi

Manuel de notation : [\[Métrique\]](#), page 58.

Fichiers d'initialisation : 'scm/time-signature-settings.scm'.

Morceaux choisis : [Section "Rythme"](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section "Auto_beam_engraver"](#) dans *Référence des propriétés internes*, [Section "Beam"](#) dans *Référence des propriétés internes*, [Section "BeamForbidEvent"](#) dans *Référence des propriétés internes*, [Section "beam-interface"](#) dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Si une partition se termine alors qu'une ligature automatique est restée inachevée, cette dernière ligature ne sera pas imprimée du tout. C'est également valable dans le cas d'une musique polyphonique saisie avec la syntaxe << ... \ \ ... >>, où une voix se terminerait sans que la dernière ligature ne soit achevée. Le plus simple, en pareil cas, est de spécifier manuellement les dernières ligatures.

Le traducteur **Timing** est par défaut affecté au contexte **Score**. Définir la métrique dans une portée aura donc des effets sur les ligatures de toutes les autres. Par voie de conséquence, la définition de la métrique apparaissant dans une autre portée annulera les aménagements précédemment apportés aux règles de ligature. Il est donc préférable, pour éviter tout désagrément, de ne spécifier la métrique que dans une seule portée.

```

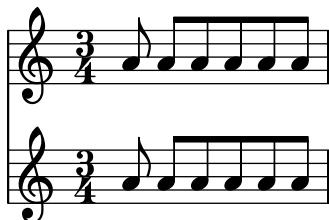
<<
\new Staff {
  \time 3/4
  \set Timing.baseMoment = #(ly:make-moment 1 8)
  \set Timing.beatStructure = #'(1 5)
  \set Timing.beamExceptions = #'()
  \repeat unfold 6 { a8 }
}
\new Staff {

```

```

\repeat unfold 6 { a8 }
}
>>

```

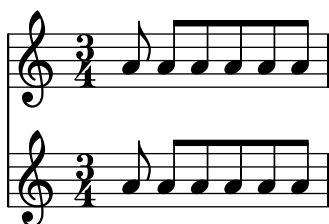


Vous pouvez adapter les règles de ligature par défaut pour une métrique particulière de telle sorte que ces règles que vous aurez définies soient toujours prises en compte. La modification des règles de ligature automatiques est abordée au chapitre [\[Métrique\]](#), page 58.

```

<<
\new Staff {
  \overrideTimeSignatureSettings
    3/4          % timeSignatureFraction
    1/8          % baseMomentFraction
    #'(1 5)      % beatStructure
    #'()         % beamExceptions
  \time 3/4
  \repeat unfold 6 { a8 }
}
\new Staff {
  \time 3/4
  \repeat unfold 6 { a8 }
}
>>

```



Barres de ligature manuelles

Dans certaines situations, il peut s'avérer nécessaire de supplanter l'algorithme de groupement automatique des notes, par exemple pour prolonger une ligature par-dessus un silence ou une barre de mesure, ou bien pour suivre le rythme des paroles plutôt que celui des notes. Le début et la fin de la ligature sont alors indiqués respectivement par [et].

```

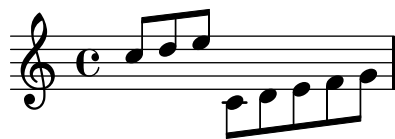
r4 r8[ g' a r] r8 g[ l a] r

```



Le positionnement des ligature manuelles se détermine comme pour toute indication attachée à une note :


```
c8^[ d e] c,_[ d e f g]
```



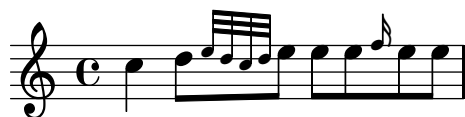
Le fait d'affubler une note particulière d'un `\noBeam` aura pour effet de l'empêcher d'être ligaturée :

```
\time 2/4
c8 c\noBeam c c
```



Notes d'ornement et normales font l'objet d'un traitement distinct. Il est donc possible de ligaturer ou non des notes d'ornement sans gêner ce qui est en place au niveau de la notation normale.

```
c4 d8[
\grace { e32[ d c d] }
e8] e[ e
\grace { f16 }
e8 e]
```



LilyPond peut déterminer automatiquement les sous-groupes à l'intérieur d'un groupement de notes, bien que le résultat ne soit pas toujours optimal. Les propriétés `stemLeftBeamCount` et `stemRightBeamCount` permettent alors d'ajuster ce comportement. Lorsque l'une ou l'autre de ces propriétés est définie, elle ne s'applique qu'une seule fois, après quoi sa définition est effacée. Dans l'exemple qui suit, le dernier `fa` n'a de ligature supplémentaire que sur sa gauche ; autrement dit, c'est la ligature à la croche qui est importante.

```
a8[ r16 f g a]
a8[ r16
\set stemLeftBeamCount = #2
\set stemRightBeamCount = #1
f16
\set stemLeftBeamCount = #1
g16 a]
```



Commandes prédéfinies

`\noBeam.`

Morceaux choisis

Crochet rectiligne et débordement de ligature

En combinant `stemLeftBeamCount`, `stemRightBeamCount` et des paires de `[]`, vous pourrez obtenir des crochets rectilignes et des ligatures qui débordent à leurs extrémités.

Pour des crochets rectilignes à droite sur des notes isolées, il suffit d'ajouter une paire d'indicateurs de ligature `[]` et de déterminer `stemLeftBeamCount` à zéro, comme dans l'exemple 1.

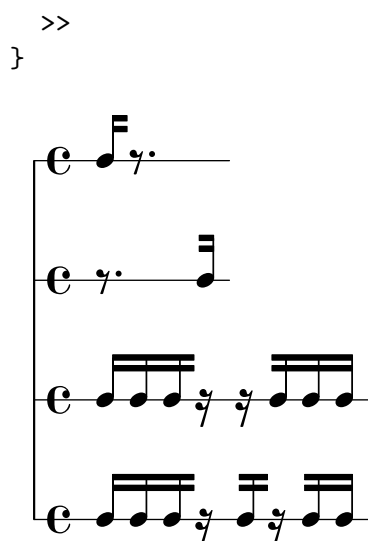
Pour des crochets rectiligne à gauche, c'est `stemRightBeamCount` qu'il faudra déterminer (exemple 2).

Pour que les barres de ligature débordent sur la droite, `stemRightBeamCount` doit avoir une valeur positive ; pour un débordement à gauche, c'est sur `stemLeftBeamCount` qu'il faut jouer. Tout ceci est illustré par l'exemple 3.

Il est parfois judicieux, lorsqu'une note est encadrée de silences, de l'affubler de crochets rectilignes de part et d'autre. L'exemple 4 montre qu'il suffit d'adjoindre à cette note un `[]`.

(Notez bien que `\set stemLeftBeamCount` sera toujours synonyme de `\once \set`. Autrement dit, la détermination des ligatures n'est pas « permanente » ; c'est la raison pour laquelle les crochets du `c'16[]` isolé du dernier exemple n'ont rien à voir avec le `\set` indiqué deux notes auparavant.)

```
\score {
  <<
    % Example 1
    \new RhythmicStaff {
      \set stemLeftBeamCount = #0
      c16[]
      r8.
    }
    % Example 2
    \new RhythmicStaff {
      r8.
      \set stemRightBeamCount = #0
      c16[]
    }
    % Example 3
    \new RhythmicStaff {
      c16 c
      \set stemRightBeamCount = #2
      c16 r r
      \set stemLeftBeamCount = #2
      c16 c c
    }
    % Example 4
    \new RhythmicStaff {
      c16 c
      \set stemRightBeamCount = #2
      c16 r
      c16[]
      r16
      \set stemLeftBeamCount = #2
      c16 c
    }
  }
```



Voir aussi

Manuel de notation : Section 5.4.2 [Direction et positionnement], page 579, [Notes d'ornement], page 102.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Beam” dans *Référence des propriétés internes*, Section “BeamEvent” dans *Référence des propriétés internes*, Section “Beam_engraver” dans *Référence des propriétés internes*, Section “beam-interface” dans *Référence des propriétés internes*, Section “Stem_engraver” dans *Référence des propriétés internes*.

Liens de croches en soufflet

Les ligatures en soufflet permettent d'indiquer qu'un petit groupe de notes se joue en accélérant ou en ralentissant sans pour autant modifier le tempo du morceau. L'étendue du soufflet s'indique par [et], et son orientation est déterminée par la propriété `grow-direction` de l'objet `Beam`.

Lorsque la sortie MIDI doit refléter les *ritardando* ou *accelerando* indiqués par une ligature en soufflet, les notes qui la composent doivent être regroupées dans une expression musicale délimitée par des accolades, précédée de la commande `\featherDurations`. Cette commande détermine le ratio entre les durées des premières et dernières notes du groupe en question.

Les crochets indiquent l'étendue de la ligature et les accolades les notes concernées par une modification de leur durée. Il s'agit en général du même groupe de notes, mais les deux commandes sont indépendantes l'une de l'autre.

Dans l'exemple ci-après, les huit doubles croches occupent exactement le même espace qu'une blanche, mais la première est moitié moins longue que la dernière et celles qui les séparent s'allongent peu à peu. Les quatre triples croches qui suivent vont s'accélérant, alors que les quatre dernières gardent un tempo régulier.

```
\override Beam #'grow-direction = #LEFT
\featherDurations #(ly:make-moment 2 1)
{ c16[ c c c c c c c c] }
\override Beam #'grow-direction = #RIGHT
\featherDurations #(ly:make-moment 2 3)
{ c32[ d e f] }
% revert to non-feathered beams
\override Beam #'grow-direction = #'()
{ g32[ a b c] }
```



Si le résultat imprimable ne reflète les durées que de manière approximative, la sortie MIDI sera quant à elle parfaitement « ponctuelle ».

Commandes prédéfinies

`\featherDurations.`

Voir aussi

Morceaux choisis : [Section “Rythme” dans *Morceaux choisis*](#).

Problèmes connus et avertissements

La commande `\featherDurations` ne permet de traiter que de très courts extraits, avec une faible amplitude.

1.2.5 Mesures

Barres de mesure

Les barres de mesures délimitent les mesures, mais peuvent aussi indiquer une reprise. En principe, elles sont insérées automatiquement en respectant la métrique en vigueur.

Il est possible de forcer l'impression d'une barre de mesure spéciale, avec la commande `\bar` – c'est d'ailleurs l'habitude en fin de morceau, où l'on voit une double barre :

```
e4 d c2 \bar "|."
```



Rien ne s'oppose à ce que la dernière note d'une mesure ne s'arrête avant la barre de mesure ; on considère simplement qu'elle se prolonge sur la mesure suivante. Des débordements à répétition finissent par générer une musique comprimée ou qui sort de la page, pour la simple et bonne raison que les sauts de ligne automatiques ne peuvent intervenir qu'à la fin d'une mesure complète, autrement dit lorsque toutes les notes sont terminées avant la fin de la mesure.

Note : Une durée erronée peut empêcher les sauts de ligne, ce qui conduit à une musique compressée, voire à un débordement de la page.

Il est possible d'autoriser un saut de ligne même s'il n'y a pas de barre de mesure visible, en utilisant :

```
\bar ""
```

Ceci insérera une barre de mesure invisible, et permettra – sans pour autant le forcer – de sauter de ligne à cet endroit, sans incrémenter le numéro de mesure. Pour forcer le saut de ligne, référez vous à [Section 4.3.1 \[Sauts de ligne\], page 510](#).

Cette barre invisible, ainsi que d'autres barres spéciales, peuvent être insérées manuellement n'importe où. Lorsqu'elles coïncident avec la fin d'une mesure, elles remplacent la simple barre que LilyPond aurait insérée automatiquement. Dans le cas contraire, la barre spécifiée s'insérera là où vous l'aurez positionnée.

Ces insertions n'affectent en rien le calcul du positionnement automatique des barres de mesure à suivre ni les propriétés y afférentes – numérotation, altérations accidentelles, sauts de ligne. . .

Lorsqu'une barre manuelle est insérée à l'endroit où viendrait se placer une barre normale, seul l'effet visuel en sera modifié.

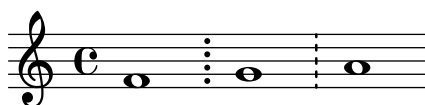
Vous disposez de deux types de barres simples et de cinq différentes doubles barres :

```
f1 \bar "|"
f1 \bar "."
g1 \bar "||"
a1 \bar ".|"
b1 \bar ".|."
c1 \bar "||."
d1 \bar "||."
e1
```



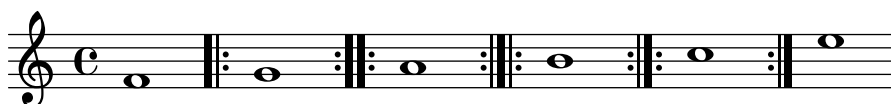
ainsi que d'une barre en pointillé et d'une discontinue :

```
f1 \bar ":"
g1 \bar "dashed"
a1
```



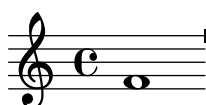
et de cinq types de barre de reprise :

```
f1 \bar ":|"
g1 \bar ":|:"
a1 \bar ":|.|:"
b1 \bar ":|.|."
c1 \bar ":|."
e1
```



De plus, une barre de mesure peut s'imprimer sous la forme d'une coche :

```
f1 \bar "'"
```



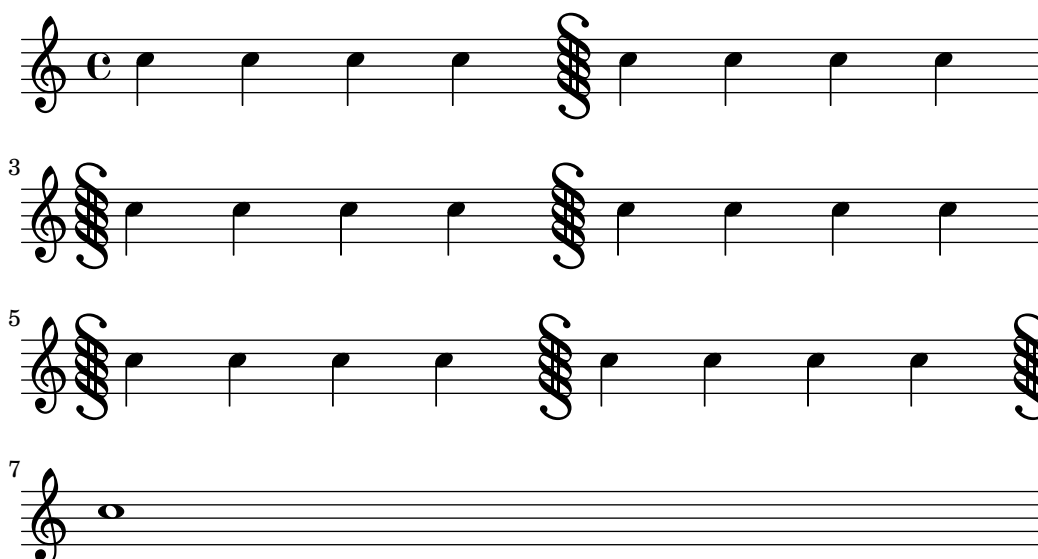
On rencontre habituellement ces signes dans le chant grégorien ; nous vous invitons dans ce cadre particulier à plutôt utiliser `\divisioMinima`, comme indiqué au paragraphe [\[Divisions\]](#), [page 421](#) du chapitre consacré au grégorien.

L'insertion d'un *segno* directement sur la portée s'obtient à l'aide de trois types de barre de mesure, comme indiqué ci-dessous :

```

c4 c c c
\bar "S"
c4 c c c \break
\bar "S"
c4 c c c
\bar "|S"
c4 c c c \break
\bar "|S"
c4 c c c
\bar "S|"
c4 c c c \break
\bar "S|"
c1

```



Bien que l'on puisse insérer des barres de reprise manuellement, LilyPond n'en déduira pas pour autant qu'il s'agit d'un passage à répéter. Il est préférable d'indiquer les passages répétés à l'aide des différentes commandes de reprise (voir [Section 1.4 \[Répétitions et reprises\]](#), page 134) qui se chargeront d'imprimer le type de barre approprié.

Dans les faits, un "`||:`" équivaut à un "`|:`" sauf s'il intervient à un saut de ligne : une double barre sera alors imprimée en fin de portée, et la barre de reprise au début de la nouvelle.

```

c4 c c c
\bar "||:"
c4 c c c \break
\bar "||:"
c4 c c c

```



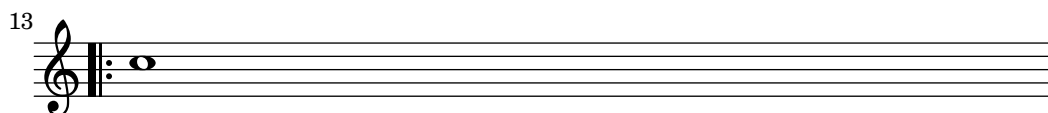
LilyPond dispose de six différents moyens de combiner une barre de reprise avec un signe *segno* :

```

c4 c c c
\bar ":|S"
c4 c c c \break
\bar ":|S"
c4 c c c
\bar ":|S."
c4 c c c \break
\bar ":|S."
c4 c c c
\bar "S|:"
c4 c c c \break
\bar "S|:"
c4 c c c
\bar ".S|:"
c4 c c c \break
\bar ".S|:"
c4 c c c
\bar ":|S|:"
c4 c c c \break
\bar ":|S|:"
c4 c c c
\bar ":|S.|:"
c4 c c c \break
\bar ":|S.|:"
c1

```

The musical notation shows six staves, each with a single measure of music. The first measure of each staff contains four quarter notes (C4, D4, E4, F4). The second measure of each staff contains four quarter notes (G4, A4, B4, C5). The staves are numbered 1 through 11 at the beginning of each line. The notation includes bar lines, repeat signs, and a final double bar line at the end of the sixth staff.



Par ailleurs, la commande `\inStaffSegno` crée une barre de mesure surmontée d'un *segno*, et sait coopérer avec l'instruction `\repeat volta`.

Dans une partition comprenant plusieurs portées, la commande `\bar` placée sur une portée s'applique automatiquement à toutes les portées. Les barres de mesure que l'on obtient alors sont d'un seul tenant sur les portées d'un `StaffGroup`, d'un `PianoStaff` ou d'un `GrandStaff`.

```
<<
  \new StaffGroup <<
    \new Staff {
      e4 d
      \bar "||"
      f4 e
    }
    \new Staff { \clef bass c4 g e g }
  >>
  \new Staff { \clef bass c2 c2 }
>>
```



La commande `'\bar type_de_barre'` sert de raccourci pour `'\set Timing.whichBar = type_de_barre'`. Dès que l'on définit `whichBar`, une barre de mesure est créée selon le style défini.

Le type de barre de mesure par défaut utilisé pour l'insertion automatique est `"|"`. Vous pouvez en changer à tout moment grâce à `'\set Timing.defaultBarType = type_de_barre'`.

Voir aussi

Manuel de notation : [Section 4.3.1 \[Sauts de ligne\]](#), page 510, [Section 1.4 \[Répétitions et reprises\]](#), page 134, [\[Regroupement de portées\]](#), page 172.

Morceaux choisis : [Section “Rythme”](#) dans *Morceaux choisis*.

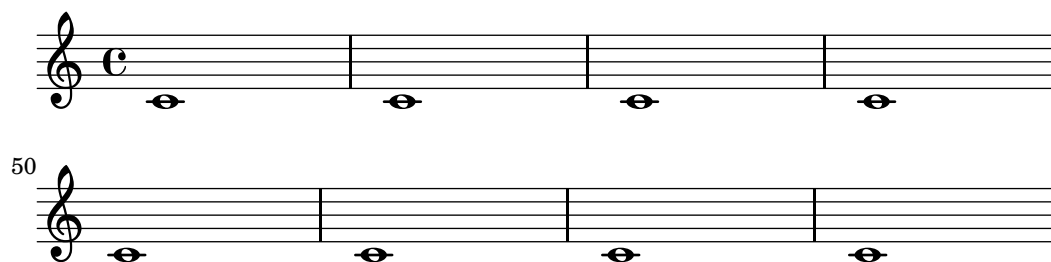
Référence des propriétés internes : [Section “BarLine”](#) dans *Référence des propriétés internes* (faisant partie du contexte `Staff`), [Section “SpanBar”](#) dans *Référence des propriétés internes* (sur plusieurs portées), [Section “Timing-translator”](#) dans *Référence des propriétés internes* (pour les propriétés liées au temps).

Numéros de mesure

Les numéros de mesure sont imprimés par défaut à chaque début de ligne, sauf la première. Ce nombre est stocké par la propriété `currentBarNumber` qui sera mise à jour à chaque mesure. Vous pouvez aussi le définir de manière arbitraire :

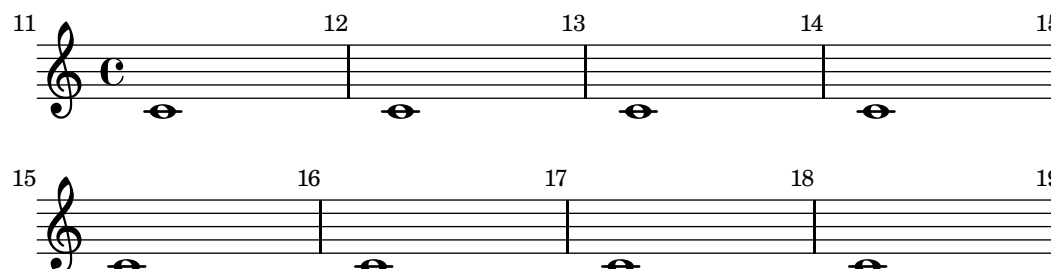
```
c1 c c c
\break
\set Score.currentBarNumber = #50
```


c1 c c c



Vous pouvez imprimer un numéro de mesure à intervalle régulier plutôt qu'en tête de chaque ligne. Pour y arriver, il faudra dans un premier temps annuler le comportement par défaut afin que les numéros puissent être imprimés ailleurs qu'en début de ligne. Tout ceci est contrôlé par la propriété `break-visibility` du `BarNumber`. Elle se compose de trois commutateurs – définis à « vrai » (`#t`) ou « faux » (`#f`) – pour spécifier si le numéro de mesure est visible ou non. Les valeurs sont rangées dans l'ordre suivant : **visible en fin de ligne**, **visible en cours de ligne** et **visible en début de ligne**. Voici comment imprimer partout les numéros de mesure :

```
\override Score.BarNumber #'break-visibility = #'#(#t #t #t)
\set Score.currentBarNumber = #11
% Permit first bar number to be printed
\bar ""
c1 | c | c | c
\break
c1 | c | c | c
```

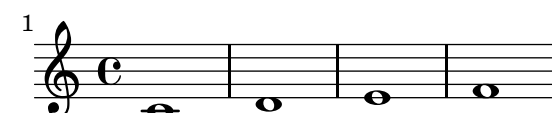


Morceaux choisis

Afficher le numéro de la première mesure

Par défaut, LilyPond n'affiche pas le premier numéro de mesure s'il est inférieur à 2. Le fait de définir `barNumberVisibility` à `all-bar-numbers-visible` vous permettra d'imprimer n'importe quel numéro pour la première mesure. Notez que l'impression d'un numéro de mesure ne peut intervenir que s'il y a une barre. Aussi, pour pouvoir le faire au début d'un morceau, devrez-vous ajouter une barre vide avant la première note.

```
\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \bar ""
  c1 | d | e | f \break
  g1 | e | d | c
}
```

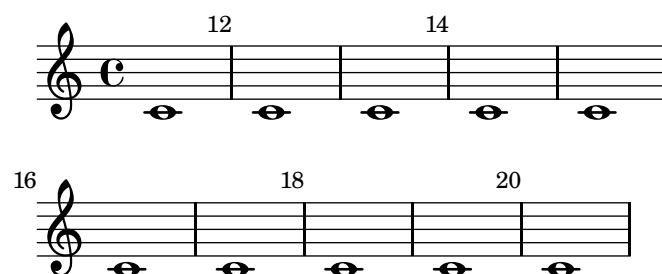




Imprimer les numéros de mesure à intervalle régulier

Vous pouvez imprimer un numéro de mesure à intervalle régulier plutôt qu'en tête de chaque ligne seulement, en recourant à la propriété `barNumberVisibility`. Voici comment afficher le numéro toutes les deux mesures sauf en fin de ligne.

```
\relative c' {
  \override Score.BarNumber #'break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = #11
  % Permit first bar number to be printed
  \bar ""
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c | c | c | c
  \break
  c1 | c | c | c | c
}
```



Inscrire le numéro de mesure dans un cadre ou un cercle

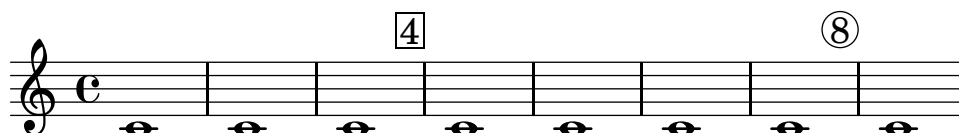
Les numéros de mesure peuvent être encadrés ou entourés d'un cercle.

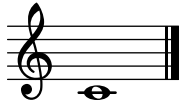
```
\relative c' {
  % Prevent bar numbers at the end of a line and permit them elsewhere
  \override Score.BarNumber #'break-visibility = #end-of-line-invisible
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2
  \override Score.BarNumber #'font-size = #2

  % Draw a box round the following bar number(s)
  \override Score.BarNumber #'stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \repeat unfold 5 { c1 }

  % Draw a circle round the following bar number(s)
  \override Score.BarNumber #'stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 4 { c1 } \bar "|."
}
```





Numérotation des mesures et alternatives

Deux méthodes alternatives vous permettent de gérer la numérotation des mesures en cas de reprises.

```
\relative c'{
  \set Score.alternativeNumberingStyle = #'numbers
  \repeat volta 3 { c4 d e f | }
  \alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
  }
  c1 \break
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \repeat volta 3 { c,4 d e f | }
  \alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
  }
  c1
}
```

Alignement des numéros de mesure

Les numéros de mesure s'alignent en principe sur la droite de l'objet dont ils dépendent. C'est normalement le coin gauche de la portée ou, en cours de ligne, à gauche de la barre. Vous pouvez toutefois les centrer par rapport à la barre ou les afficher à droite de la barre.

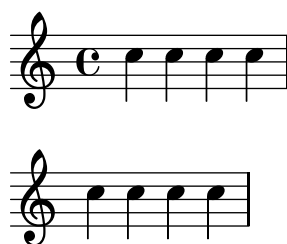
```
\relative c' {
  \set Score.currentBarNumber = #111
  \override Score.BarNumber #'break-visibility = #all-visible
  % Increase the size of the bar number by 2
  \override Score.BarNumber #'font-size = #2
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c1
  % Center-align bar numbers
  \override Score.BarNumber #'self-alignment-X = #CENTER
  c1 | c1
  % Left-align bar numbers
  \override Score.BarNumber #'self-alignment-X = #LEFT
  c1 | c1
}
```

*Suppression des numéros de mesure d'une partition*

Désactiver le graveur concerné – `Bar_number_engraver` – donnera une partition – contexte `Score` – sans numéros de mesure.

```
\layout {
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}

\relative c'' {
  c4 c c c \break
  c4 c c c
}
```

**Voir aussi**

Morceaux choisis : [Section “Rythme”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “BarNumber”](#) dans *Référence des propriétés internes*, [Section “Bar_number_engraver”](#) dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les numéros de mesure peuvent entrer en collision avec les crochets d'un [Section "StaffGroup"](#) dans *Référence des propriétés internes*. La propriété `padding` – décalage – de l'objet [Section "BarNumber"](#) dans *Référence des propriétés internes* permet alors d'ajuster leur positionnement.

Vérification des limites et numéros de mesure

Les tests de limite de mesure (ou tests de mesure) aident à détecter les erreurs dans les durées. Un test de mesure s'écrit avec une barre verticale, `|`. Lors du traitement, elle doit correspondre à une barre de mesure. Sinon, un avertissement est émis qui indique le numéro de ligne où est détectée l'erreur. Dans l'exemple suivant, le deuxième test de mesure signale une erreur.

```
\time 3/4 c2 e4 | g2 |
```

Le test de mesure peut être aussi utilisé dans les paroles, par exemple :

```
\lyricmode {
  \time 2/4
  Twin -- kle | Twin -- kle
}
```

Des durées incorrectes font échouer les tests de mesure et peuvent souvent mettre la partition sens dessus dessous, particulièrement s'il s'agit de musique polyphonique. Vérifier les tests de mesure qui ont échoué et les durées incorrectes est un bon moyen de commencer à corriger sa partition.

Lorsque plusieurs tests successifs présentent un même décalage, seul le message d'avertissement concernant la première occurrence est affiché. L'origine du problème est de fait plus évidente.

Il est aussi possible d'attribuer une autre valeur au symbole `|`, en assignant une expression musicale à `pipeSymbol`. Dans l'exemple suivant, le `|` servira à insérer une double barre là où il apparaît, au lieu de simplement vérifier que la fin de la mesure est atteinte.

```
pipeSymbol = \bar "||"
{
  c'2 c' |
  c'2 c'
  c'2 | c'
  c'2 c'
}
```



Lorsque l'on recopie de longues pièces, il peut être utile de vérifier que les numéros de mesure de LilyPond correspondent à l'original que l'on recopie. Cela se fait avec `\barNumberCheck`. Par exemple,

```
\barNumberCheck #123
```

affiche un avertissement lors du traitement si le numéro de mesure à ce point (variable `currentBarNumber`) n'est pas 123.

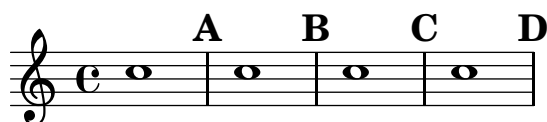
Voir aussi

Morceaux choisis : [Section "Rythme"](#) dans *Morceaux choisis*.

Indications de repère

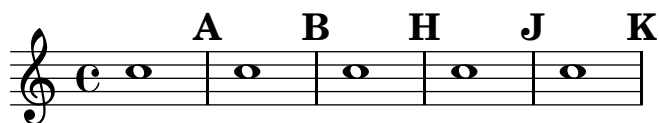
Indiquer un repère s'obtient grâce à la commande `\mark`.

```
c1 \mark \default
c1 \mark \default
c1 \mark \default
c1 \mark \default
```



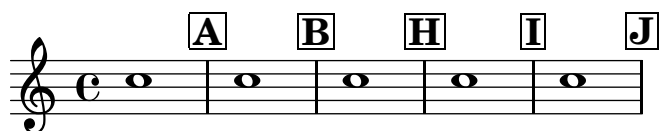
Lorsque vous utilisez `\mark \default`, le repère s'incrémente automatiquement ; toutefois donner un nombre en argument permet de spécifier manuellement le repère en question. La valeur à utiliser est enregistrée dans la propriété `rehearsalMark`.

```
c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
```



La lettre **I** n'est pas utilisée, conformément aux usages de la gravure. Cependant, vous pourrez intégrer la lettre **I** en utilisant l'une des commandes suivantes selon que ce repère doit être simple, inclus dans un rectangle ou dans un cercle :

```
\set Score.markFormatter = #format-mark-alphabet
\set Score.markFormatter = #format-mark-box-alphabet
\set Score.markFormatter = #format-mark-circle-alphabet
\set Score.markFormatter = #format-mark-box-alphabet
c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
```



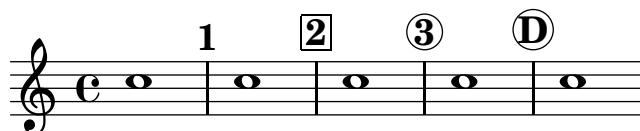
Le style de repère est déterminé par la propriété `markFormatter`. Il s'agit d'une fonction qui prend en arguments le repère en cours (un entier) ainsi que le contexte en cours, et retournera un objet de type *markup*. Dans l'exemple qui suit, `markFormatter` est réglé pour une procédure type. Quelques mesures plus loin, son comportement est modifié pour imprimer un repère encadré.

```
\set Score.markFormatter = #format-mark-numbers
c1 \mark \default
c1 \mark \default
\set Score.markFormatter = #format-mark-box-numbers
```

```

c1 \mark \default
\set Score.markFormatter = #format-mark-circle-numbers
c1 \mark \default
\set Score.markFormatter = #format-mark-circle-letters
c1

```



Le fichier ‘scm/translation-functions.scm’ comporte les définitions de `format-mark-numbers` (comportement par défaut), `format-mark-box-numbers`, `format-mark-letters` et `format-mark-box-letters`. Vous pouvez vous en inspirer pour d’autres fonctions de formatage.

`format-mark-barnumbers`, `format-mark-box-barnumbers` et `format-mark-circle-barnumbers` permettent d’imprimer le numéro de mesure au lieu des compteurs alphabétique ou numérique.

On peut aussi spécifier manuellement une marque de repère :

```
\mark "A1"
```

`Score.markFormatter` sera sans effet sur des repères ainsi définis. Un `\markup` peut néanmoins s’utiliser en argument.

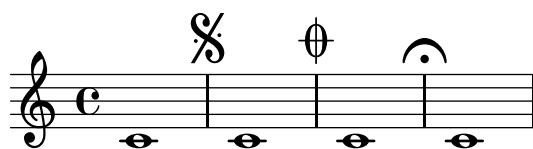
```
\mark \markup{ \box A1 }
```

Un `\mark` peut contenir un glyphe musical tel que le signe *segno*.

```

c1 \mark \markup { \musicglyph #"scripts.segno" }
c1 \mark \markup { \musicglyph #"scripts.coda" }
c1 \mark \markup { \musicglyph #"scripts.ufermata" }
c1

```



Pour connaître les différents symboles accessibles par `\musicglyph`, consultez [Section A.8 \[La fonte Feta\]](#), page 620.

Pour affiner le positionnement des repères, veuillez vous référer à [Section 1.8.2 \[Mise en forme du texte\]](#), page 221, et tout particulièrement ce qui concerne la `break-alignable-interface` au chapitre [Section 5.5.1 \[Alignement des objets\]](#), page 593.

Les définitions `format-mark-numbers` et `format-mark-letters` sont inscrites dans le fichier ‘scm/translation-functions.scm’. Elles seront une source d’inspiration en matière de fonctions de formatage.

Voir aussi

Manuel de notation : [Section A.8 \[La fonte Feta\]](#), page 620, [Section 1.8.2 \[Mise en forme du texte\]](#), page 221, [Section 5.5.1 \[Alignement des objets\]](#), page 593.

Fichiers d’initialisation : ‘scm/translation-functions.scm’.

Morceaux choisis : [Section “Rythme”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “MarkEvent”](#) dans *Référence des propriétés internes*, [Section “Mark-engraver”](#) dans *Référence des propriétés internes*, [Section “RehearsalMark”](#) dans *Référence des propriétés internes*.

1.2.6 Fonctionnalités rythmiques particulières

Notes d'ornement

Les petites notes sont des ornements entièrement écrits. Leur taille est un peu plus petite que celle des notes normales et elles n'occupent pas de temps dans la mesure.

```
c4 \grace b16 a4(  
\grace { b16[ c16] } a2)
```



Les plus courantes sont les acciaccatures, qui doivent se jouer très vite, et qui s'écrivent sous forme d'une petite note barrée (sur la hampe) et liée. L'appoggiature est une petite note non barrée, qui vole une fraction à la durée de la note réelle qui la suit. LilyPond dispose aussi, grâce à la fonction `\slashedGrace`, d'une petite note barrée et dépourvue de liaison, qui viendra s'insérer entre deux notes déjà liées.

```
\acciaccatura d8 c4  
\appoggiatura e8 d4  
\acciaccatura { g16[ f] } e2  
\slashedGrace a,8 g4  
\slashedGrace b16 a4(  
\slashedGrace b8 a2)
```



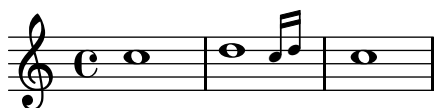
Les petites notes se placent de façon synchrone entre les différentes portées. Dans l'exemple suivant, il y a deux petites double-croches pour chaque petite croche.

```
<<  
  \new Staff { e2 \grace { c16[ d e f] } e2 }  
  \new Staff { c2 \grace { g8[ b] } c2 }  
>>
```



La commande `\afterGrace` sert à placer une petite note après une note réelle – et non *avant* comme d'ordinaire. Cette commande requiert deux arguments : la note réelle, et la ou les petites notes qui s'y rattachent.

```
c1 \afterGrace d1 { c16[ d] } c1
```



Les petites notes se placent alors aux trois quarts de la durée de la note réelle. Cette fraction peut être changée en définissant `afterGraceFraction`. Dans l'exemple suivant, vous pouvez observer la différence entre le comportement par défaut, à 15/16 et enfin à la moitié de la durée de base.

```
<<
\new Staff {
  c1 \afterGrace d1 { c16[ d] } c1
}
\new Staff {
  #(define afterGraceFraction (cons 15 16))
  c1 \afterGrace d1 { c16[ d] } c1
}
\new Staff {
  #(define afterGraceFraction (cons 1 2))
  c1 \afterGrace d1 { c16[ d] } c1
}
>>
```



Des silences invisibles permettent d'ajuster l'espace entre les petites notes et la note réelle. Nous pourrions positionner ces petites notes à sept huitièmes de la durée de la note de base :

```
\new Voice {
  <<
    { d1~\trill_( }
    { s2 s4. \grace { c16[ d] } }
  >>
  c1)
}
```



Les expressions `\grace` obéissent à des règles typographiques particulières, notamment en matière d'orientation et de taille des objets. De ce fait, toute subtilité de mise en forme devra être indiquée **à l'intérieur** de l'expression introduite par `\grace` ; ces réglages additionnels doivent également être désactivés dans cette même expression.

```
\new Voice {
  \acciaccatura {
    \stemDown
    f16->
    \stemNeutral
```

```

}
g4 e c2
}

```



Morceaux choisis

Utilisation de hampe barrée pour une note normale

Le trait que l'on trouve sur les hampes des acciaccatures peut être appliqué dans d'autres situations.

```

\relative c'' {
  \override Flag #'stroke-style = #"grace"
  c8( d2) e8( f4)
}

```



Mise en forme des notes d'ornement

Il est possible de changer globalement la mise en forme des notes d'ornement dans un morceau, au moyen des fonctions `add-grace-property` et `remove-grace-property`. Ici, par exemple, on ôte la définition de l'orientation des objets `Stem` pour toutes les petites notes, afin que les hampes ne soient pas toujours orientées vers le haut, et on leur préfère des têtes en forme de croix.

```

\relative c'' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
      \acciaccatura { f16 } g4
      \grace { d16[ e ] } f4
      \appoggiatura { f,32[ g a ] } e2
    }
  }
}

```



Redéfinition des réglages de mise en forme par défaut des notes d'ornement

Vous pouvez modifier les valeurs des variables `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`, `stopAcciaccaturaMusic`, `startAppoggiaturaMusic` et `stopAppoggiaturaMusic` afin d'en personnaliser les effets. Pour plus de détails, voir le fichier `'ly/grace-init.ly'`.

```

startAcciaccaturaMusic = {
  <>(
    \override Flag #'stroke-style = #"grace"

```

```

\slurDashed
}

stopAcciaccaturaMusic = {
  \revert Flag #'stroke-style
  \slurSolid
  <>)
}

\relative c'' {
  \acciaccatura d8 c1
}

```



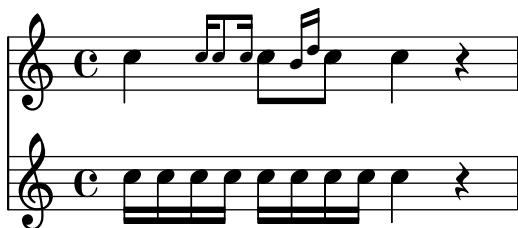
Positionnement des notes d'ornement avec espace flottant

Lorsqu'est activée la propriété '**strict-grace-spacing**', l'espacement des notes d'ornement se fera de manière « élastique ». Autrement dit, elles seront décollées de leur note de rattachement : LilyPond commence par espacer les notes normales, puis les ornements sont placés à la gauche de leur note de rattachement.

```

\relative c'' {
  <<
    \override Score.SpacingSpanner #'strict-grace-spacing = ##t
    \new Staff \new Voice {
      \afterGrace c4 { c16[ c8 c16] }
      c8[ \grace { b16[ d] } c8]
      c4 r
    }
    \new Staff {
      c16 c c c c c c c c4 r
    }
  >>
}

```



Voir aussi

Glossaire musicologique : Section “ornements” dans *Glossaire*, Section “acciaccature” dans *Glossaire*, Section “appogiature” dans *Glossaire*.

Fichiers d'initialisation : ‘ly/grace-init.ly’.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Référence des propriétés internes : Section “GraceMusic” dans *Référence des propriétés internes*, Section “Grace_beam_engraver” dans *Référence des propriétés internes*, Section

“Grace_engraver” dans *Référence des propriétés internes*, Section “Grace_spacing_engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Un groupe de notes ligaturées constituant une *acciaccatura* apparaîtra comme une *appoggiatura*, c’est-à-dire sans trait.

La synchronisation des petites notes se fait de façon parfois surprenante, car les autres objets de la portée – barre de mesure, armure, etc. – sont eux aussi synchrones. Pensez-y lorsque vous mêlez des portées comprenant des petites notes et d’autres sans :

```
<<
\new Staff { e4 \bar "|:" \grace c16 d2. }
\new Staff { c4 \bar "|:" d2. }
>>
```



Il est possible de remédier à cela en insérant, sur les autres portées, des silences invisibles dans une expression précédée de `\grace` et correspondant à la durée des petites notes :

```
<<
\new Staff { e4 \bar "|:" \grace c16 d2. }
\new Staff { c4 \bar "|:" \grace s16 d2. }
>>
```



Lorsque des petites notes font l’objet d’un contexte de voix, cela perturbe la présentation des voix normales. Vous pourrez y remédier en insérant un silence ou une note entre la commande affectant la voix et les petites notes.

```
accMusic = {
  \acciaccatura { f8 } e8 r8 \acciaccatura { f8 } e8 r4
}

\new Staff {
  <<
    \new Voice {
      \relative c'' {
        r8 r8 \voiceOne \accMusic \oneVoice r8 |
        r8 \voiceOne r8 \accMusic \oneVoice r8 |
      }
    }
  }
  \new Voice {
```

```

\relative c' {
  s8 s8 \voiceTwo \accMusic \oneVoice s8 |
  s8 \voiceTwo r8 \accMusic \oneVoice s8 |
}
}
>>
}

```



Seules des expressions musicales séquentielles peuvent être utilisées pour des petites notes ; il n'est pas possible d'imbriquer ni de juxtaposer des sections de petites notes, faute de quoi le traitement du code peut échouer ou produire des erreurs.

En ce qui concerne la sortie MIDI, les petites notes ont une durée du quart de la valeur que vous leur attribuez. Par voie de conséquence, si la durée globale d'une succession de petites notes venait à dépasser la durée de la note qui précède, vous déclencheriez une erreur du type « *Going back in MIDI time* ». Il vous faudra donc raccourcir les petites notes. Par exemple,

```
\acciaccatura { c'8[ d' e' f' g'] }
```

deviendrait :

```
\acciaccatura { c'16[ d' e' f' g'] }
```

ou bien modifier explicitement l'échelle des durées :

```
\acciaccatura { \scaleDurations 1/2 { c'8[ d' e' f' g'] } }
```

Voir [\[Changement d'échelle des durées\]](#), page 46.

Alignement et cadences

Dans un contexte orchestral, une cadence constitue un problème spécifique. Lors du montage d'une partition contenant une cadence, tous les autres instruments doivent sauter autant de notes que ce qu'en comporte la cadence, faute de quoi ils démarreraient trop tôt ou trop tard.

Les fonctions `mmrest-of-length` ou `skip-of-length` permettent de résoudre ce problème. Ces fonctions Scheme prennent en argument un fragment de musique, et génèrent un `\skip` ou un silence multimesures d'une durée correspondant à ce fragment.

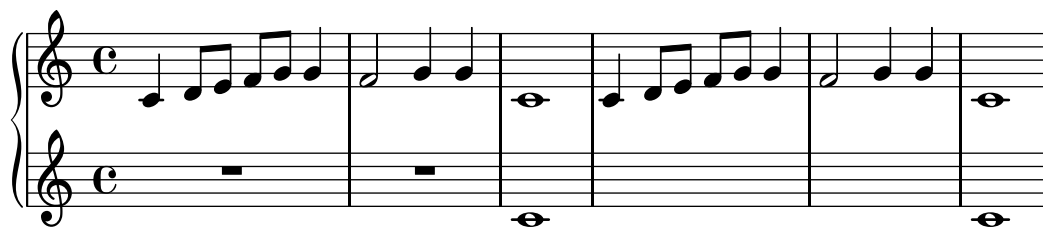
```

MyCadenza = \relative c' {
  c4 d8 e f g g4
  f2 g4 g
}

\new GrandStaff <<
  \new Staff {
    \MyCadenza c'1
    \MyCadenza c'1
  }
  \new Staff {
    #(mmrest-of-length MyCadenza)
    c'1
    #(skip-of-length MyCadenza)
    c'1
  }
}

```

>>



Voir aussi

Glossaire musicologique : [Section “cadenza” dans *Glossaire*](#).

Morceaux choisis : [Section “Rythme” dans *Morceaux choisis*](#).

Gestion du temps

Le temps est administré par le `Timing_translator`, qui réside en principe dans le contexte `Score`. Un alias, `Timing`, sera ajouté au contexte auquel le `Timing_translator` est rattaché. Déclarer explicitement un contexte `Voice` ou `Staff` assure l’existence de cet alias.

`Timing` dispose des propriétés suivantes afin de garder trace du minutage de la partition.

`currentBarNumber`

Le numéro de la mesure en cours. Un exemple d’utilisation se trouve au chapitre [\[Numéros de mesure\]](#), page 94.

`measureLength`

La longueur de la mesure, dans la métrique en cours. Pour une mesure à 4/4, elle est de 1, et de 3/4 pour une mesure à 6/8. Sa valeur détermine où peut s’insérer une barre et comment seront générées les ligatures automatiques.

`measurePosition`

Le moment où l’on en est dans la mesure en cours. Cette quantité est remise à 0 dès lors qu’on dépasse `measureLength` ; la variable `currentBarNumber` est alors incrémentée.

`timing` Lorsqu’on lui assigne la valeur *vrai*, les valeurs ci-dessus mentionnées sont mises à jour à chaque pas. Fixée à *faux*, le graveur restera indéfiniment dans la mesure en cours.

Le calage peut être modifié en réglant explicitement l’une de ces variables. Dans l’exemple qui suit, nous réglons la métrique à 4/4, tout en fixant `measureLength` à 5/4. Arrivé à 4/8 dans la troisième mesure, nous avançons de 1/8, en assignant 5/8 à `measurePosition`, raccourcissant donc cette mesure d’une croche. La barre de mesure suivante tombera donc à 9/8 et non à 5/8.

```
\new Voice \relative c' {
  \set Timing.measureLength = #(ly:make-moment 5 4)
  c1 c4 |
  c1 c4 |
  c4 c
  \set Timing.measurePosition = #(ly:make-moment 5 8)
  b4 b b8 |
  c4 c1 |
}
```



Comme le montre cet exemple, `ly:make-moment n d` construit une durée de n/d fois une ronde. Par conséquent, `ly:make-moment 1 8` correspond à une croche, et `ly:make-moment 7 16` à la durée de sept doubles croches.

Voir aussi

Manuel de notation : [Numéros de mesure], page 94, [Musique sans métrique], page 67

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Références des propriétés internes : Section “Timing_translator” dans *Référence des propriétés internes*, Section “Score” dans *Référence des propriétés internes*

1.3 Signes d’interprétation

RONDO
Allegro

Ce chapitre traite des différentes indications d’interprétation que l’on peut trouver sur les partitions.

1.3.1 Signes d’interprétation attachés à des notes

Nous allons voir au cours de ces lignes comment ajouter aux notes des indications d’interprétation – articulation, ornementation, nuance – et aborderons la manière de générer vos propres signes.

Articulations et ornements

Les différents symboles qui indiquent des ponctuations ou des modes de jeu différents s’ajoutent aux notes de la manière suivante :

`note\nom`

Les valeurs de *nom* sont répertoriées dans l’annexe Section A.13 [Liste des signes d’articulation], page 687. En voici un exemple :

```
c4\staccato c\mordent b2\turn
c1\fermata
```



Certains signes d’articulation disposent d’un raccourci. On les ajoute à chaque note au moyen d’un tiret suivi du caractère correspondant à l’articulation désirée. C’est entre autres le cas pour *marcato*, *stopped*, *tenuto*, *staccatissimo*, *accent*, *staccato*, et *portato*, comme l’illustre l’exemple ci-dessous.

```
c4-^ c-+ c-- c-|
c4-> c-. c2-_
```



Même si LilyPond place automatiquement ces symboles, selon les règles contenues dans le fichier ‘`scm/script.scm`’, il est possible de l’obliger à les positionner au-dessus ou en dessous de la note, comme indiqué au chapitre [Section 5.4.2 \[Direction et positionnement\]](#), page 579.

Les articulations sont des objets de type `script` ; les propriétés de ces objets sont abordées plus en détail au chapitre [Section “Script”](#) dans *Référence des propriétés internes*.

Une articulation peut se rattacher aussi bien à un silence qu’à une note, mais **pas à un silence multimesures**. Il existe cependant un cas particulier : le point d’orgue – ou point d’arrêt – attaché à un silence valant l’intégralité de la mesure. LilyPond dispose à cet effet de la commande `\fermataMarkup`, qui créera un objet `MultiMeasureRestText` rattaché à ce « silence multimesures ».

```
\override Script #'color = #red
\override MultiMeasureRestText #'color = #blue
a2\fermata r\fermata
R1\fermataMarkup
```



En dehors des articulations habituelles, vous pouvez adjoindre du texte – avec ou sans mise en forme – à n’importe quelle note. Voir à ce propos [\[Commentaires textuels\]](#), page 213.

Pour plus d’information sur la manière d’ordonner `Scripts` et `TextScripts`, consultez le chapitre [Section “Positionnement des objets”](#) dans *Manuel d’initiation*.

Morceaux choisis

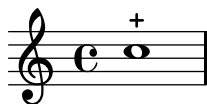
Modification de la signification des raccourcis pour les signes d’articulation

Les raccourcis sont répertoriés dans le fichier ‘`ly/script-init.ly`’, dans lequel on retrouve les variables `dashHat`, `dashPlus`, `dashDash`, `dashBar`, `dashLarger`, `dashDot`, et `dashUnderscore` ainsi que leur valeur par défaut. Ces valeurs peuvent être modifiées selon vos besoins. Il suffit par exemple, pour affecter au raccourci `-+` (`dashPlus`) le symbole du trille en lieu et place du `+` (caractère plus), d’assigner la valeur `trill` à la variable `dashPlus` :


```
\relative c'' { c1-+ }
```

```
dashPlus = "trill"
```

```
\relative c'' { c1-+ }
```



Contrôle de l'ordre vertical des articulations et ornements

Les symboles s'ordonnent verticalement suivant la propriété **script-priority**. Plus sa valeur numérique est faible, plus le symbole sera proche de la note. Dans l'exemple suivant, l'objet **TextScript** – le dièse – a d'abord la propriété la plus basse, et il est donc placé plus près de la note ; ensuite c'est l'objet **Script** – le mordant – qui a la propriété la plus basse, et il se place alors sous le dièse. Lorsque deux objets ont la même priorité, c'est l'ordre dans lequel ils sont indiqués qui détermine lequel sera placé en premier.

```
\relative c''' {
  \once \override TextScript #'script-priority = #-100
  a2^\prall^\markup { \sharp }

  \once \override Script #'script-priority = #-100
  a2^\prall^\markup { \sharp }
}
```



Création d'un grupetto retardé

Obtenir un *grupetto* retardé et dans lequel la note la plus basse est altérée requiert quelques surcharges. La propriété **outside-staff-priority** doit être désactivée (**#f**) pour éviter qu'elle prenne le pas sur la propriété **avoid-slur**. La valeur assignée à **halign** permet de gérer horizontalement le « retard ».

```
\relative c'' {
  \once \override TextScript #'avoid-slur = #'inside
  \once \override TextScript #'outside-staff-priority = #f
  c2(\markup \tiny \override #'(baseline-skip . 1) {
    \halign #-4
    \center-column {
      \sharp
      \musicglyph #"scripts.turn"
    }
  })
}
```

}



Voir aussi

Glossaire musicologique : Section “tenuto” dans *Glossaire*, Section “accent” dans *Glossaire*, Section “staccato” dans *Glossaire*, Section “portato” dans *Glossaire*.

Manuel d’initiation : Section “Positionnement des objets” dans *Manuel d’initiation*.

Manuel de notation : [Commentaires textuels], page 213, Section 5.4.2 [Direction et positionnement], page 579, Section A.13 [Liste des signes d’articulation], page 687, [Trilles], page 132.

Fichiers d’initialisation : ‘scm/script.scm’.

Morceaux choisis : Section “Signes d’interprétation” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Script” dans *Référence des propriétés internes*, Section “TextScript” dans *Référence des propriétés internes*.

Nuances

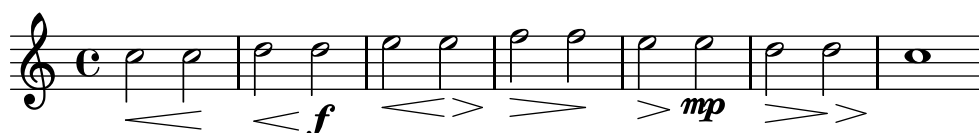
À chaque nuance absolue correspond une commande qui peut être indiquée après une note : `c4\ff` par exemple. Les commandes de nuance disponibles sont `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\ffffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz` et `\rfz`. Les nuances se placent aussi bien en dessous qu’au-dessus de la portée ; pour plus d’information, consultez Section 5.4.2 [Direction et positionnement], page 579.

```
c2\ppp c\mp
c2\rfz c^\mf
c2_\spp c^\ff
```



Un crescendo est délimité par `\<` et `\!`, ou peut se terminer par une commande de nuance explicite, ou bien un decrescendo ou un nouveau crescendo. Il en va de même pour un diminuendo. Au lieu de `\<` et `\>`, vous pouvez utiliser `\cr` et `\decr`, auquel cas LilyPond n’imprimera pas de soufflet (*hairpin* en anglais).

```
c2\< c\!
d2\< d\f
e2\< e\>
f2\> f\!
e2\> e\mp
d2\> d\>
c1\!
```



Un soufflet terminé par un simple `\!` prendra fin sur la droite de la tête de note à laquelle il est attaché. Dans le cas où il se termine par l'intervention d'un autre soufflet (contraire ou non), il prendra fin au milieu de la tête de note affublée d'un `\<` ou d'un `\>`, et le nouveau soufflet débutera à l'extrémité droite de cette même tête de note.

```
c1\< | c4 a c\< a | c4 a c\! a\< | c4 a c a\!
```



Il en va de même lorsqu'un soufflet est interrompu par une nuance explicite. Notez bien que la largeur occupée par cette nuance explicite influe sur la terminaison du soufflet :

```
c1\< | c4 a c\mf a | c1\< | c4 a c\ffff a
```



Les indications de nuance sont attachées aux notes ; aussi, lorsque l'on veut faire se succéder plusieurs nuances pendant une note tenue, il faudra avoir recours à des silences invisibles :

```
c4\< c\! d\> e\!  
<< f1 { s4 s4\< s4\> s4\! } >>
```



On peut avoir recours à l'indication `\espressivo` pour indiquer un crescendo suivi d'un decrescendo sur une même note. Gardez à l'esprit qu'il s'agit d'une articulation, et en aucun cas d'une nuance.

```
c2 b4 a  
g1\espressivo
```



La commande `\cresc` permet d'indiquer textuellement le début d'un crescendo. `\decresc` ou `\dim` marquent le début d'un decrescendo. Les lignes d'extension sont gérées automatiquement.

```
g8\cresc a b c b c d e\mf |  
f8\decresc e d c e\> d c b |  
a1\dim ~ |  
a2. r4\! |
```



Une indication textuelle peut indiquer, au lieu d'un soufflet, un changement de nuance :

```

\crescTextCresc
c4\< d e f\! |
\dimTextDecresc
g4\> e d c\! |
\dimTextDecr
e4\> d c b\! |
\dimTextDim
d4\> c b a\! |
\crescHairpin
\dimHairpin
c4\< d\! e\> d\! |

```



Pour créer des indications de nuance qui restent alignées avec les nuances habituelles, reportez-vous au chapitre [\[Personnalisation des indications de nuance\]](#), page 117.

Le positionnement vertical des nuances est géré par le [Section “DynamicLineSpanner”](#) dans [Référence des propriétés internes](#).

L'utilisation d'un contexte `Dynamics` permet de graver les nuances sur leur propre ligne – Il suffit de placer des silences invisibles pour gérer le temps. Bien que le contexte `Dynamics` accepte des notes pour indiquer les durées, celles-ci ne seront pas imprimées. Le contexte `Dynamics` peut aussi contenir des indications textuelles avec ou sans extenseur, ainsi que les indications de pédale.

```

<<
\new Staff \relative c' {
  c2 d4 e |
  c4 e e,2 |
  g'4 a g a |
  c1 |
}
\new Dynamics {
  s1\< |
  s1\f |
  s2\dim s2-"rit." |
  s1\p |
}
>>

```



Commandes prédéfinies

```

\dynamicUp, \dynamicDown, \dynamicNeutral, \crescTextCresc, \dimTextDim,
\dimTextDecr, \dimTextDecresc, \crescHairpin, \dimHairpin.

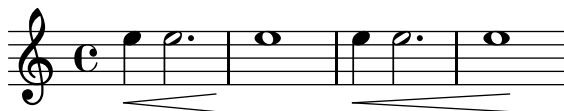
```

Morceaux choisis

Soufflets et barres de mesure

En principe, un soufflet – (de)crescendo imprimé sous forme graphique – commence au bord gauche de la note de départ, et se termine au bord droit de la note d'arrivée. Cependant, si la note d'arrivée est sur un premier temps, le soufflet s'arrêtera au niveau de la barre de mesure qui la précède. Ce comportement peut être annulé en assignant *faux* (#f) à la propriété 'to-barline.

```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin #'to-barline = ##f
  e4\< e2.
  e1\!
}
```



Ajustement de la longueur d'un soufflet

Si un soufflet est trop court, il suffit d'ajuster la propriété `minimum-length` de l'objet `Hairpin` pour l'allonger.

```
\relative c'' {
  c4\< c\! d\> e\!
  \override Hairpin #'minimum-length = #5
  << f1 { s4 s\< s\> s\! } >>
}
```



Impression de soufflets « al niente »

Des crescendos ou decrescendos *al niente* peuvent être indiqués de manière graphique, en assignant *vrai* (#t) à la propriété `circled-tip`, ce qui affiche un cercle à leur extrémité.

```
\relative c'' {
  \override Hairpin #'circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```



Alignement vertical des nuances indications textuelles

Vous pourrez, en jouant sur la propriété 'Y-extent, aligner les différents objets `DynamicLineSpanner` (soufflets ou textuels) quelle que soit leur étendue, par rapport à un même point de référence. Tous les éléments seront alors centrés sur une même ligne, ce qui sera visuellement plus agréable.

C'est le même principe qui sert à aligner les indications textuelles sur une ligne de référence.

```

music = \relative c' {
  a'2\p b\f
  e4\p f\f> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner #'staff-padding = #2.0
  \override DynamicLineSpanner #'Y-extent = #'(-1.5 . 1.5)
  \override TextScript #'Y-extent = #'(-1.5 . 1.5)
  \music
}

```



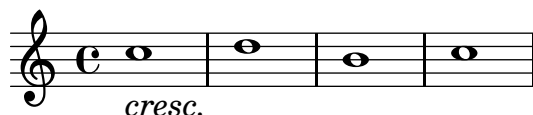
Masquage de l'extension des nuances textuelles

Les crescendos et decrescendos indiqués textuellement – tels que *cresc.* ou *dim.* – sont suivis de pointillés qui montrent leur étendue. On peut empêcher l'impression de ces pointillés.

```

\relative c'' {
  \override DynamicTextSpanner #'style = #'none
  \crescTextCresc
  c1\< | d | b | c\!
}

```



Modification du texte et de l'extension de nuances textuelles

Le texte par défaut des crescendos et decrescendos se change en modifiant les propriétés de contexte `crescendoText` et `decrescendoText`. L'aspect de la ligne d'extension est fonction de la propriété `'style` du `DynamicTextSpanner`. Sa valeur par défaut est `'hairpin`, mais d'autres valeurs sont disponibles, comme `'line`, `'dashed-line` et `'dotted-line`.

```

\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner #'style = #'dotted-line

```

```

a2\< a
a2 a
a2 a
a2 a\mf
}

```



Voir aussi

Glossaire musicologique : Section “al niente” dans *Glossaire*, Section “crescendo” dans *Glossaire*, Section “decrescendo” dans *Glossaire*, Section “soufflet” dans *Glossaire*.

Manuel d’initiation : Section “Articulations et nuances” dans *Manuel d’initiation*.

Manuel de notation : Section 5.4.2 [Direction et positionnement], page 579, [Personnalisation des indications de nuance], page 117, Section 3.5.3 [Contenu de la sortie MIDI], page 488, Section 3.5.5 [Gestion des nuances en MIDI], page 489.

Morceaux choisis : Section “Signes d’interprétation” dans *Morceaux choisis*.

Référence des propriétés internes : Section “DynamicText” dans *Référence des propriétés internes*, Section “Hairpin” dans *Référence des propriétés internes*, Section “DynamicLineSpanner” dans *Référence des propriétés internes*, Section “Dynamics” dans *Référence des propriétés internes*.

Personnalisation des indications de nuance

La manière la plus simple de personnaliser une indication de nuance consiste à utiliser un objet `\markup`.

```
moltoF = \markup { molto \dynamic f }
```

```

\relative c' {
  <d e>16_\moltoF <d e>
  <d e>2..
}

```

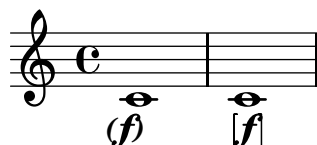


Vous pouvez créer des indications de nuance éditoriales (entre parenthèses ou crochets) grâce aux étiquettes (*mode markup*) ; la syntaxe en est abordée au chapitre [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

```

roundF = \markup {
  \center-align \concat { \bold { \italic ( }
    \dynamic f \bold { \italic ) } } }
boxF = \markup { \bracket { \dynamic f } }
\relative c' {
  c1_\roundF
  c1_\boxF
}

```



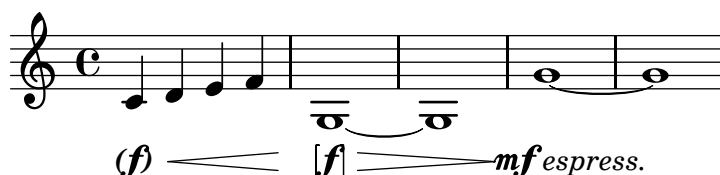
Grâce à la fonction `make-dynamic-script`, vous pouvez créer de nouvelles marques textuelles que vous combinerez avec les signes de nuance.

```
sfzp = #(make-dynamic-script "sfzp")
\relative c' {
  c4 c c\sfpz c
}
```



`make-dynamic-script` accepte en argument tout objet de type *markup*. Notez bien que la police des nuances ne contient que les caractères *f*, *m*, *p*, *r*, *s* et *z*, et que les marques de nuance possèdent des propriétés particulières et prédéfinies quant à leur police. Aussi, lorsque vous créez du texte en pareille situation, nous vous recommandons d'utiliser `\normal-text` pour annuler ces propriétés. L'intérêt majeur de recourir à la fonction `make-dynamic-script` plutôt qu'à un simple *markup* réside dans l'assurance que ces objets personnalisés et les soufflets seront alignés lorsqu'attachés à une même note.

```
roundF = \markup { \center-align \concat {
  \normal-text { \bold { \italic ( } }
  \dynamic f
  \normal-text { \bold { \italic ) } } } }
boxF = \markup { \bracket { \dynamic f } }
mfEspress = \markup { \center-align \line {
  \hspace #3.7 mf \normal-text \italic espress. } }
roundFdynamic = #(make-dynamic-script roundF)
boxFdynamic = #(make-dynamic-script boxF)
mfEspressDynamic = #(make-dynamic-script mfEspress)
\relative c' {
  c4_\roundFdynamic\< d e f
  g,1~_\boxFdynamic\>
  g
  g'\mfEspressDynamic
  g
}
```

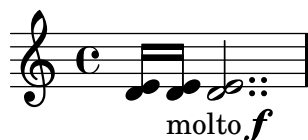


La construction d'une indication de nuance personnalisée peut aussi se faire en langage Scheme ; voir [Section "Construction d'un markup en Scheme"](#) dans *Extension de LilyPond* pour en connaître les modalités.

```
moltoF = #(make-dynamic-script
  (markup #:normal-text "molto"
    #:dynamic "f"))
```



```
\relative c' {
  <d e>16 <d e>
  <d e>2..\moltoF
}
```



L'utilisation des fontes en mode *markup* est abordée au chapitre [Sélection de la fonte et de la taille], page 223.

Voir aussi

Manuel de notation : Section 1.8.2 [Mise en forme du texte], page 221, [Sélection de la fonte et de la taille], page 223, Section 3.5.3 [Contenu de la sortie MIDI], page 488, Section 3.5.5 [Gestion des nuances en MIDI], page 489.

Morceaux choisis : Section “Signes d’interprétation” dans *Morceaux choisis*.

Manuel d’extension : Section “Construction d’un markup en Scheme” dans *Extension de LilyPond*.

1.3.2 Signes d’interprétation sous forme de courbe

Ce chapitre traite des signes d’interprétation imprimés sous forme de courbe : liaisons d’articulation ou de phrasé, respirations, chutes et sauts.

Liaisons d’articulation

Une liaison d’articulation indique que les notes doivent être jouées liées, ou *legato*. Ces liaisons s’indiquent au moyen de parenthèses.

Note : Lorsque la musique est polyphonique, la liaison doit se terminer dans la voix où elle a été entamée.

```
f4( g a) a8 b(
a4 g2 f4)
<c e>2( <b d>2)
```



Vous pouvez décider de l’orientation des liaisons par rapport à la portée, comme indiqué au chapitre Section 5.4.2 [Direction et positionnement], page 579.

Une seule liaison d’articulation peut être imprimée à la fois. S’il est nécessaire d’imprimer une liaison plus longue, englobant des liaisons plus courtes, utilisez des [Liaisons de phrasé], page 122.

Une liaison est par défaut dessinée d’un trait plein. Il est aussi possible de l’imprimer sous la forme de tirets ou en pointillé :

```
c4( e g2)
\slurDashed
g4( e c2)
```

```
\slurDotted
c4( e g2)
\slurSolid
g4( e c2)
```



En utilisant `\slurHalfDashed`, la première moitié de la liaison aura un trait discontinu et continu pour la seconde. L'inverse s'obtient avec `\slurHalfSolid`.

```
c4( e g2)
\slurHalfDashed
g4( e c2)
\slurHalfSolid
c4( e g2)
\slurSolid
g4( e c2)
```



Vous pouvez même personnaliser la densité des tirets d'une liaison :

```
c4( e g2)
\slurDashPattern #0.7 #0.75
g4( e c2)
\slurDashPattern #0.5 #2.0
c4( e g2)
\slurSolid
g4( e c2)
```



Commandes prédéfinies

`\slurUp`, `\slurDown`, `\slurNeutral`, `\slurDashed`, `\slurDotted`, `\slurHalfDashed`, `\slurHalfSolid`, `\slurDashPattern`, `\slurSolid`.

Morceaux choisis

Accords et double liaison d'articulation

Certains auteurs utilisent deux liaisons lorsqu'ils veulent lier des accords. Dans LilyPond, il faut pour cela activer la propriété `doubleSlurs`.

```
\relative c' {
  \set doubleSlurs = ##t
  <c e>4( <d f> <c e> <d f>)
}
```



Positionnement d'une annotation à l'intérieur d'une liaison

Lorsqu'une annotation doit s'inscrire à l'intérieur d'une liaison, la propriété `outside-staff-priority` doit être désactivée.

```
\relative c' {
  \override TextScript #'avoid-slur = #'inside
  \override TextScript #'outside-staff-priority = ##f
  c2(~\markup { \halign #-10 \natural } d4.) c8
}
```



Dessin d'une liaison d'articulation au trait discontinu

Grâce à la propriété `dash-definition`, une liaison d'articulation peut être formée de traits discontinus variables. `dash-definition` se compose d'une liste de `segments-discontinus` (*dash-elements*). Chaque `segment-discontinus` contient une liste de paramètres qui déterminent le comportement du trait pour une section de la liaison.

Cette liaison se définit selon le paramètre de Bézier `t` qui est compris entre 0 (l'extrémité gauche de la liaison) et 1 (l'extrémité droite de la liaison). Chaque `segment-discontinus` se composera selon la liste (`t-début t-fin segment-style segment-taille`). La portion de liaison allant de `t-début` à `t-fin` aura un trait `segment-style` de longueur `segment-taille`. `segment-taille` est exprimé en espace de portée ; un `segment-style` à 1 donnera un trait plein.

```
\relative c' {
  \once \override
    Slur #'dash-definition = #'((0 0.3 0.1 0.75)
                                (0.3 0.6 1 1)
                                (0.65 1.0 0.4 0.75))

  c4( d e f)
  \once \override
    Slur #'dash-definition = #'((0 0.25 1 1)
                                (0.3 0.7 0.4 0.75)
                                (0.75 1.0 1 1))

  c4( d e f)
}
```



Voir aussi

Glossaire musicologique : Section “liaison” dans *Glossaire*.

Manuel d'initiation : Section “Non-imbrication des crochets et liaisons” dans *Manuel d'initiation*.

Manuel de notation : Section 5.4.2 [Direction et positionnement], page 579, [Liaisons de phrasé], page 122.

Morceaux choisis : [Section “Signes d’interprétation” dans *Morceaux choisis*.](#)

Référence des propriétés internes : [Section “Slur” dans *Référence des propriétés internes*.](#)

Liaisons de phrasé

Une liaison de phrasé relie plusieurs notes en délimitant une phrase musicale. On indique les points de départ et d’arrivée avec `\(` et `\)` respectivement.

```
c4\( d( e) f(
e2) d\)
```



D’un point de vue typographique, rien ne distingue une liaison de phrasé d’une liaison d’articulation. Cependant, LilyPond les considère comme des objets différents. Une commande `\slurUp` n’affectera donc pas une liaison de phrasé. Vous pouvez décider de l’orientation des liaisons de phrasé par rapport à la portée, comme indiqué au chapitre [Section 5.4.2 \[Direction et positionnement\]](#), page 579.

Il n’est pas possible d’avoir plusieurs liaisons de phrasé en même temps.

Une liaison est par défaut dessinée d’un trait plein. Il est aussi possible de l’imprimer sous la forme de tirets ou en pointillé :

```
c4\( e g2\)
\phrasingSlurDashed
g4\( e c2\)
\phrasingSlurDotted
c4\( e g2\)
\phrasingSlurSolid
g4\( e c2\)
```



En utilisant `\phrasingslurHalfDashed`, la première moitié de la liaison aura un trait discontinu et continu pour la seconde. L’inverse s’obtient avec `\phrasingslurHalfSolid`.

```
c4\( e g2\)
\phrasingSlurHalfDashed
g4\( e c2\)
\phrasingSlurHalfSolid
c4\( e g2\)
\phrasingSlurSolid
g4\( e c2\)
```



Vous pouvez même personnaliser la densité des tirets d’une liaison :

```

c4\ ( e g2\ )
\phrasingSlurDashPattern #0.7 #0.75
g4\ ( e c2\ )
\phrasingSlurDashPattern #0.5 #2.0
c4\ ( e g2\ )
\phrasingSlurSolid
g4\ ( e c2\ )

```



La personnalisation des lignes discontinues est identique pour les liaisons de phrasé et les liaisons d'articulation. Pour plus de détails, référez-vous aux morceaux choisis de la section [\[Liaisons d'articulation\]](#), page 119.

Commandes prédéfinies

```

\phrasingSlurUp, \phrasingSlurDown, \phrasingSlurNeutral, \phrasingSlurDashed,
\phrasingSlurDotted, \phrasingSlurHalfDashed, \phrasingSlurHalfSolid,
\phrasingSlurDashPattern, \phrasingSlurSolid.

```

Voir aussi

Manuel d'initiation : [Section “Non-imbrication des crochets et liaisons”](#) dans *Manuel d'initiation*.

Manuel de notation : [Section 5.4.2 \[Direction et positionnement\]](#), page 579, [\[Liaisons d'articulation\]](#), page 119.

Morceaux choisis : [Section “Signes d'interprétation”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “PhrasingSlur”](#) dans *Référence des propriétés internes*.

Signes de respiration

Les indications de respiration sont indiquées par la commande `\breathe`.

```
c2. \breathe d4
```



Un signe de respiration interrompt obligatoirement les ligatures, même automatiques. Pour passer outre ce fonctionnement, voir [\[Barres de ligature manuelles\]](#), page 86.

```
c8 \breathe d e f g2
```



LilyPond gère les *divisiones*, signes utilisés en notation ancienne pour indiquer les respirations. Pour de plus amples détails, voir [\[Divisions\]](#), page 421.

Morceaux choisis

Modification de l'indicateur de respiration

On peut choisir le glyphe imprimé par cette commande, en modifiant la propriété `text` de l'objet `BreathingSign`, pour lui affecter n'importe quelle indication textuelle.

```
\relative c' ' {
  c2
  \override BreathingSign #'text =
    \markup { \musicglyph #"scripts.rvarcomma" }
  \breathe
  d2
}
```



Insertion d'une césure

Une surcharge de la propriété `'text` de l'objet `BreathingSign` permet de créer une marque de césure. LilyPond dispose également d'une variante courbée.

```
\relative c' ' {
  \override BreathingSign #'text = \markup {
    \musicglyph #"scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4

  \override BreathingSign #'text = \markup {
    \musicglyph #"scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}
```



Voir aussi

Glossaire musicologique : [Section “césure” dans *Glossaire*](#).

Manuel de notation : [\[Divisions\]](#), page 421.

Morceaux choisis : [Section “Signes d’interprétation” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “BreathingEvent” dans *Référence des propriétés internes*](#), [Section “BreathingSign” dans *Référence des propriétés internes*](#), [Section “Breathing-sign-engraver” dans *Référence des propriétés internes*](#).

Chutes et sauts

Des indications de désinence peuvent être obtenues au moyen de la commande `\bendAfter`. Leur direction s’indique au moyen des signes plus (vers le haut) ou moins (vers le bas). Le chiffre indique l’intervalle avec la note de départ.

```
c2\bendAfter #+4
c2\bendAfter #-4
```

```

c2\bendAfter #+6.5
c2\bendAfter #-6.5
c2\bendAfter #+8
c2\bendAfter #-8

```



Morceaux choisis

Ajustement du galbe des chutes ou sauts

La propriété `shortest-duration-space` peut devoir être retouchée pour ajuster l'apparence des chutes ou sauts.

```

\relative c'' {
  \override Score.SpacingSpanner #'shortest-duration-space = #4.0
  c2-\bendAfter #5
  c2-\bendAfter #-4.75
  c2-\bendAfter #8.5
  c2-\bendAfter #-6
}

```



Voir aussi

Glossaire musical : [Section “chute” dans *Glossaire*](#), [Section “saut” dans *Glossaire*](#).

Morceaux choisis : [Section “Signes d’interprétation” dans *Morceaux choisis*](#).

1.3.3 Signes d’interprétation sous forme de ligne

Cette partie traite de la manière de générer des signes d’interprétation d’aspect linéaire, tels les glissandos, arpèges et trilles.

Glissando

Un glissando relie une hauteur à une autre en passant par chaque hauteur intermédiaire. On l’obtient en accolant la commande `\glissando` à la première note.

```

g2\glissando g'
c2\glissando c,
\afterGrace f,1\glissando f'16

```



Un glissando peut intervenir au moment d’un changement de portée :

```

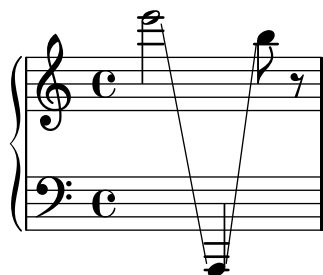
\new PianoStaff <<
  \new Staff = "right" {
    e''2\glissando

```

```

\change Staff = "left"
a,,4\glissando
\change Staff = "right"
b''8 r |
}
\new Staff = "left" {
  \clef bass
  s1
}
>>

```

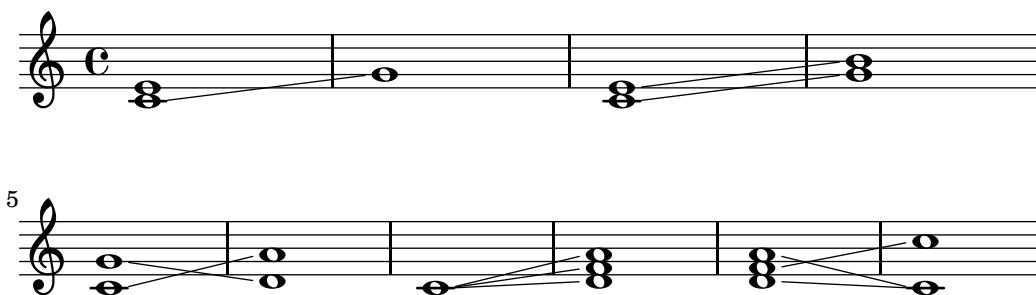


Un glissando peut affecter des notes d'un accord. En dehors du cas où les notes des deux accords sont reliées directement l'une à l'autre, les relations s'établissent à l'aide de la commande `\glissandoMap` ; les notes d'un accord sont numérotées à partir de zéro et dans leur ordre d'apparition dans le fichier `'ly'`.

```

<c, e>1\glissando g' |
<c, e>1\glissando |
<g' b> |
\break
\set glissandoMap = #'((0 . 1) (1 . 0))
<c, g'>1\glissando |
<d a'> |
\set glissandoMap = #'((0 . 0) (0 . 1) (0 . 2))
c1\glissando |
<d f a> |
\set glissandoMap = #'((2 . 0) (1 . 0) (0 . 1))
<f d a'>1\glissando |
<c c'> |

```



Un glissando est indiqué graphiquement, par une ligne ou des vaguelettes – voir [Section 5.4.7 \[Styles de ligne\]](#), page 591.

Morceaux choisis

Glissando contemporain

De nos jours, il peut arriver que la note d'arrivée d'un glissando soit absente de la partition. Pour ce faire, il vous faudra utiliser une cadence et « masquer » la note d'arrivée.

```
\relative c'' {
  \time 3/4
  \override Glissando #'style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
}
```



Ajout de marques temporelles à un long glissando

Lorsqu'un glissando s'étend dans la durée, on trouve parfois des indications temporelles, matérialisées par des hampes sans tête de note. De telles hampes permettent aussi d'indiquer des éléments intermédiaires.

L'alignement des hampes avec la ligne de glissando peut requérir quelques aménagements.

```
glissandoSkipOn = {
  \override NoteColumn #'glissando-skip = ##t
  \override NoteHead #'transparent = ##t
  \override NoteHead #'no-ledgers = ##t
}
```

```
glissandoSkipOff = {
  \revert NoteColumn #'glissando-skip
  \revert NoteHead #'transparent
  \revert NoteHead #'no-ledgers
}
```

```
\relative c'' {
  r8 f8\glissando
  \glissandoSkipOn
  f4 g a a8\noBeam
  \glissandoSkipOff
  a8
```

```

  r8 f8\glissando
  \glissandoSkipOn
  g4 a8
  \glissandoSkipOff
```

```

a8 |

r4 f\glissando \<
\glissandoSkipOn
a4\f \>
\glissandoSkipOff
b8\! r |
}

```



Saut de ligne et glissando

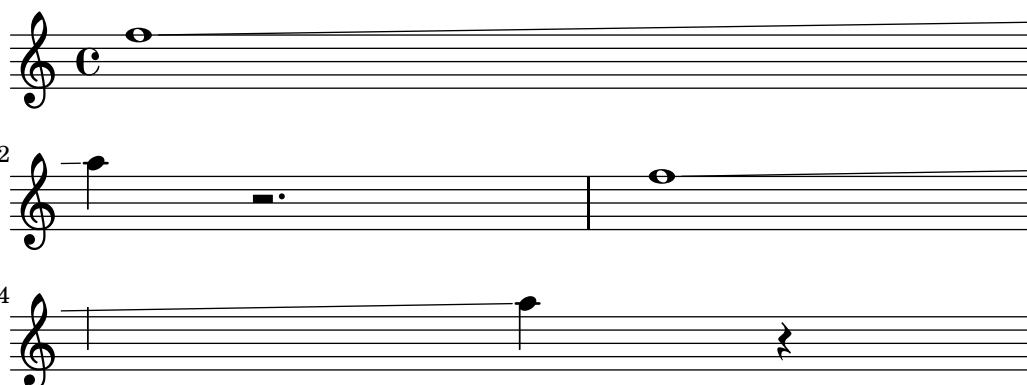
L'affectation de la valeur `#t` à la propriété `breakable`, combinée à `after-line-breaking`, permet la rupture d'une indication de glissando lors d'un saut de ligne.

```

glissandoSkipOn = {
  \override NoteColumn #'glissando-skip = ##t
  \override NoteHead #'transparent = ##t
  \override NoteHead #'no-ledgers = ##t
}

\relative c'' {
  \override Glissando #'breakable = ##t
  \override Glissando #'after-line-breaking = ##t
  f1\glissando |
  \break
  a4 r2. |
  f1\glissando
  \once \glissandoSkipOn
  \break
  a2 a4 r4 |
}

```



Voir aussi

Glossaire musicologique : [Section “glissando” dans *Glossaire*](#).

Manuel de notation : [Section 5.4.7 \[Styles de ligne\], page 591](#).

Morceaux choisis : [Section “Signes d’interprétation” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “Glissando” dans *Référence des propriétés internes*](#).

Problèmes connus et avertissements

Il n'est pas possible d'imprimer un texte (tel que *gliss.*) le long de la ligne de glissando.

Arpèges

On peut indiquer qu'un accord doit être arpégé en lui accolant la commande `\arpeggio` :

```
<c e g c>1\arpeggio
```



LilyPond dispose de différents graphismes pour indiquer un arpège ; `\arpeggioNormal` reviendra au style par défaut.

```
<c e g c>2\arpeggio
```

```
\arpeggioArrowUp
```

```
<c e g c>2\arpeggio
```

```
\arpeggioArrowDown
```

```
<c e g c>2\arpeggio
```

```
\arpeggioNormal
```

```
<c e g c>2\arpeggio
```



Des crochets indiquent que l'accord devra être plaqué et non arpégé :

```
<c e g c>2
```

```
\arpeggioBracket
```

```
<c e g c>2\arpeggio
```

```
\arpeggioParenthesis
```

```
<c e g c>2\arpeggio
```

```
\arpeggioParenthesisDashed
```

```
<c e g c>2\arpeggio
```

```
\arpeggioNormal
```

```
<c e g c>2\arpeggio
```



Les indications d'arpège peuvent se présenter sous la forme de ligne discontinue à l'aide de la propriété `'dash-details`. Pour plus de détails à ce propos, consultez [\[Liaisons d'articulation\]](#), page 119.

Un arpège peut parfois s'écrire de manière explicite, à l'aide de liaisons de tenue. Pour plus d'information, voir [\[Liaisons de prolongation\]](#), page 47.

Commandes prédéfinies

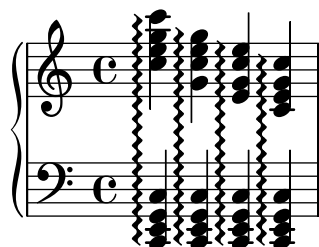
`\arpeggio`, `\arpeggioArrowUp`, `\arpeggioArrowDown`, `\arpeggioNormal`, `\arpeggioBracket`, `\arpeggioParenthesis` `\arpeggioParenthesisDashed`.

Morceaux choisis

Arpège distribué sur une partition pour piano

Dans une double portée pour piano (`PianoStaff`), un arpège peut s'étendre sur les deux portées grâce à la propriété `PianoStaff.connectArpeggios`.

```
\new PianoStaff \relative c' ' <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    <c e g c>4\arpeggio
    <g c e g>4\arpeggio
    <e g c e>4\arpeggio
    <c e g c>4\arpeggio
  }
  \new Staff {
    \clef bass
    \repeat unfold 4 {
      <c,, e g c>4\arpeggio
    }
  }
>>
```



Arpège distribué pour un autre contexte que le piano

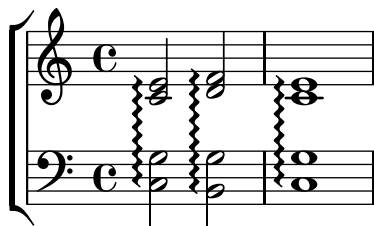
Il est possible de distribuer un arpège sur plusieurs portées d'un système autre que le `PianoStaff` dès lors que vous incluez le `Span_arpeggio_engraver` au contexte `Score`.

```
\score {
  \new ChoirStaff {
    \set Score.connectArpeggios = ##t
    <<
      \new Voice \relative c' {
        <c e>2\arpeggio
        <d f>2\arpeggio
        <c e>1\arpeggio
      }
      \new Voice \relative c {
        \clef bass
        <c g'>2\arpeggio
        <b g'>2\arpeggio
        <c g'>1\arpeggio
      }
    >>
  }
>>
```

```

    }
    \layout {
      \context {
        \Score
        \consists "Span_arpeggio_engraver"
      }
    }
  }
}

```



Arpège distribué sur plusieurs voix

Affecter le graveur `Span_arpeggio_engraver` au contexte de la portée (`Staff`) permet de distribuer un arpège sur plusieurs voix :

```

\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}

```



Voir aussi

Glossaire musicologique : [Section “arpeggio”](#) dans *Glossaire*.

Manuel de notation : [\[Liaisons d’articulation\]](#), page 119, [\[Liaisons de prolongation\]](#), page 47.

Morceaux choisis : [Section “Signes d’interprétation”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Arpeggio”](#) dans *Référence des propriétés internes*, [Section “Slur”](#) dans *Référence des propriétés internes*, [Section “PianoStaff”](#) dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Il est impossible de mêler au même instant, dans un contexte `PianoStaff`, des lignes d’arpèges connectées et d’autres non connectées.

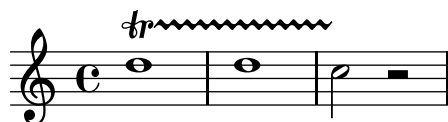
Il n’est pas possible d’imprimer des lignes d’arpège connectées sous forme de parenthèse.

Trilles

Les trilles brefs s'indiquent comme n'importe quelle ponctuation, avec un simple `\trill` ; voir [Articulations et ornements], page 109.

Les trilles plus longs sont délimités par `\startTrillSpan` et `\stopTrillSpan` :

```
d1\startTrillSpan
d1
c2\stopTrillSpan r2
```



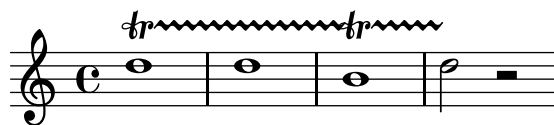
Lorsqu'un saut de ligne intervient alors qu'une prolongation de trille est présente, l'indication de trille et sa prolongation sont rappelées sur la première note de la nouvelle ligne :

```
d1\startTrillSpan
\break
d1
c2\stopTrillSpan r2
```



Lorsque des trilles interviennent sur une succession de hauteurs différentes, point n'est besoin d'explicitement la commande `\stopTrillSpan` puisque l'apparition d'un nouveau trille interrompt de fait celui qui le précédait :

```
d1\startTrillSpan
d1
b1\startTrillSpan
d2\stopTrillSpan r2
```



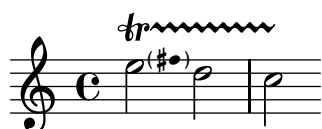
Dans l'exemple suivant, un trille se combine avec des notes d'ornement. La syntaxe d'une telle construction ainsi que le moyen de positionner les notes d'ornement avec précision est expliquée au chapitre [Notes d'ornement], page 102.

```
d1~\afterGrace
d1\startTrillSpan { c32[ d]\stopTrillSpan }
e2 r2
```



Les trilles qui font intervenir une hauteur précise peuvent être indiqués par la commande `\pitchedTrill`. Le premier argument est la note réelle ; le second est une hauteur qui sera imprimée comme une tête de note noire entre parenthèses.

```
\pitchedTrill
e2\startTrillSpan fis
d2 c2\stopTrillSpan
```



Dans l'exemple suivant, le second trille de la deuxième mesure est ambigu – le fa qui est forcé n'est pas diésé. Pour lever toute ambiguïté, il suffit de forcer l'impression de l'altération en ajoutant un ! à la note considérée.

```
\pitchedTrill
eis4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan cis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis!
eis4\stopTrillSpan
```



Commandes prédéfinies

`\startTrillSpan`, `\stopTrillSpan`.

Voir aussi

Glossaire musicologique : [Section “trille” dans *Glossaire*](#).

Manuel de notation : [\[Articulations et ornements\]](#), page 109, [\[Notes d'ornement\]](#), page 102.

Morceaux choisis : [Section “Signes d'interprétation” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “TrillSpanner” dans *Référence des propriétés internes*](#).

1.4 Répétitions et reprises



La répétition est une notion essentielle en musique, et il existe de nombreuses façons de mettre en œuvre et noter ce concept. LilyPond prend en charge les types de répétition suivants :

- volta** Le passage répété n'est pas développé, mais il est encadré par des barres de reprise et peut se terminer par plusieurs fins alternatives – ou *volte* – imprimées de gauche à droite sous des crochets. Lorsque la répétition commence au début de la pièce, aucune barre de reprise n'est gravée au début de la partition. Il s'agit de la notation courante des reprises avec fins alternatives.
- unfold** La musique répétée est développée dans la partition autant de fois qu'indiqué. Ceci est particulièrement utile pour de la musique répétitive.
- percent** Des barres obliques ou signes de pourcentage indiquent la répétition de temps ou de mesures.
- tremolo** Ce type permet de réaliser des trémolos sous forme de liens de croches.

1.4.1 Répétition d'un long passage

Cette section présente la syntaxe des répétitions longues – c'est-à-dire plusieurs mesures. Ces répétitions peuvent prendre deux formes : encadrées par des barres de reprises, ou bien développées dans la partition. Les barres et autres signes de reprise peuvent être contrôlés manuellement.

Répétitions courantes

On peut indiquer une répétition de la façon suivante :

```
\repeat volta nombre_de_fois expression_musicale
```

où *expression_musicale* représente ce qui doit être répété.

Les reprises courantes, sans alternative, s'indiquent comme ceci :

```
\repeat volta 2 { c4 d e f }
c2 d
\repeat volta 2 { d4 e f g }
```



On peut ajouter une fin alternative à l'aide de la commande `\alternative`. Chaque *alternative* est une expression musicale en elle-même ; il faudra donc les regrouper par des accolades.

```
\repeat volta nombre_de_fois expression_musicale
\alternative {
  { expression_musicale }
}
```

Si l'on donne trop peu d'alternatives en regard du nombre de fois où le passage doit être rejoué, la première alternative sera jouée plusieurs fois.

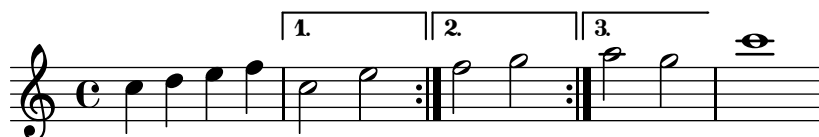
Voici une simple reprise avec une fin alternative :

```
\repeat volta 2 { c4 d e f | }
\alternative {
  { c2 e | }
  { f2 g | }
}
c1
```



Et une répétition avec plusieurs alternatives :

```
\repeat volta 3 { c4 d e f | }
\alternative {
  { c2 e | }
  { f2 g | }
  { a2 g | }
}
c1
```



Note : Lorsqu'il y a plus d'une alternative, prenez garde à ce qu'aucun caractère n'apparaisse entre l'accolade fermant une alternative et l'accolade ouvrant la suivante, au risque de ne pas obtenir le nombre voulu d'alternatives.

Note : Une clause `\relative` ne doit jamais se trouver à l'intérieur d'une section `\repeat` : vous aurez inmanquablement des portées parasites. Voir Section "Apparition d'une portée supplémentaire" dans *Utilisation des programmes*.

Lorsqu'une reprise sans fin alternative débute au milieu d'une mesure, elle devrait se terminer aussi au milieu d'une mesure, de telle sorte que les mesures soient complètes. En pareil cas, les indications de reprise ne constituent pas des barres de mesure à proprement parler ; il n'est donc pas nécessaire de faire appel à la commande `\partial` ou à des contrôles d'intégrité de mesure.

```
% no \partial here
c4 e g % no bar check here
% no \partial here
\repeat volta 4 {
  e4 |
  c2 e |
  % no \partial here
  g4 g g % no bar check here
}
% no \partial here
g4 |
a2 a |
g1 |
```



Il est possible de créer des reprises en début de morceau avec une levée. Le cas est similaire à ce que nous venons de voir. Toutefois, l'utilisation d'un `\partial` est ici nécessaire pour respecter cette entame.

```
\partial 4 % required
\repeat volta 4 {
  e4 |
  c2 e |
  % no \partial here
  g4 g g % no bar check here
}
% no \partial here
g4 |
a2 a |
g1 |
```



Lorsqu'une reprise débute par une mesure incomplète et a des fins alternatives, il est indispensable d'ajuster manuellement la propriété `Timing.measureLength` selon les préceptes suivants :

- au début de chacune des mesures incomplètes du bloc `\alternative`, ce qui est le cas en principe pour chaque fin d'alternative sauf – la plupart du temps – pour la dernière.
- au début de chaque alternative à l'exception de la première.

```
\partial 4
\repeat volta 2 { e4 | c2 e | }
\alternative {
  {
    f2 d |
    \set Timing.measureLength = #(ly:make-moment 3 4)
    g4 g g % optional bar check is allowed here
  }
  {
    \set Timing.measureLength = #(ly:make-moment 4 4)
    a2 a |
  }
}
g1 |
```



La propriété `measureLength` est abordée plus en détail au chapitre [\[Gestion du temps\]](#), page 108.

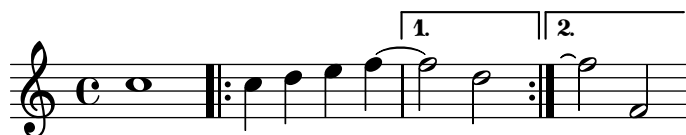
La commande `\inStaffSegno` permet d'adjoindre un symbole de *segno* à une commande `\repeat volta`. Des symboles alternatifs de barre de mesure peuvent être accessibles, dans un contexte `Score`, à l'aide des propriétés `segnoType`, `startRepeatSegnoType`, `endRepeatSegnoType` ou `doubleRepeatSegnoType` selon les besoins.

```
e1
\repeat volta 2 {
  \inStaffSegno
  f2 g a b
}
c1_"D.S." \bar "|."
```



Des liaisons de tenue peuvent être ajoutées à toute alternative :

```
c1
\repeat volta 2 { c4 d e f ~ }
\alternative {
  { f2 d }
  { f2\repeatTie f, }
}
```



Morceaux choisis

Diminution de la taille du crochet d'alternative

Les crochets indiquant les fins alternatives s'étalent tout au long de celles-ci. On peut les raccourcir en jouant sur la propriété `voltaSpannerDuration`. Dans l'exemple suivant, le crochet ne se prolonge que sur une mesure à 3/4.

```
\relative c' {
  \time 3/4
  c4 c c
  \set Score.voltaSpannerDuration = #(ly:make-moment 3 4)
  \repeat volta 5 { d4 d d }
  \alternative {
    {
      e4 e e
      f4 f f
    }
    { g4 g g }
  }
}
```



Ajout du crochet de reprise à d'autres portées

D'ordinaire, le graveur `Volta_engraver` réside dans le contexte `Score` ; les crochets précédant une reprise s'impriment donc seulement au-dessus de la portée du haut. On peut ajuster cela en déplaçant ce graveur vers les contextes de portée (`Staff`) qui doivent comporter ces crochets.

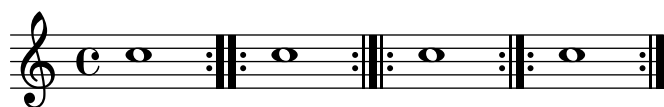
```
<<
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff \with { \consists "Volta_engraver" } { c'2 g' e' a' }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
>>
```



Succession de reprise et style de barre par défaut

LilyPond dispose de trois différents styles de barre pour indiquer une succession de reprises. Vous devez opter pour un style par défaut, à l'aide de la propriété `doubleRepeatType`.

```
\relative c'' {
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|.|:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|..:"
  \repeat volta 1 { c1 }
}
```

*Numérotation des mesures et alternatives*

Deux méthodes alternatives vous permettent de gérer la numérotation des mesures en cas de reprises.

```
\relative c'{
  \set Score.alternativeNumberingStyle = #'numbers
  \repeat volta 3 { c4 d e f | }
  \alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
  }
  c1 \break
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \repeat volta 3 { c,4 d e f | }
  \alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
  }
  c1
}
```





Voir aussi

Glossaire musicologique : Section “répétition” dans *Glossaire*, Section “volta” dans *Glossaire*.

Manuel de notation : [Barres de mesure], page 90, Section 5.1.4 [Modification des greffons de contexte], page 555, [Gestion du temps], page 108.

Morceaux choisis : Section “Répétitions” dans *Morceaux choisis*.

Référence des propriétés internes : Section “VoltaBracket” dans *Référence des propriétés internes*, Section “RepeatedMusic” dans *Référence des propriétés internes*, Section “VoltaRepeatedMusic” dans *Référence des propriétés internes*, Section “UnfoldedRepeatedMusic” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

L’extension d’une liaison à partir d’un bloc `\repeat` sur un bloc `\alternative` n’est possible que pour la première alternative. Selon le même principe, une liaison ne saurait partir de la fin d’une alternative pour se terminer au début de la reprise.

Le développement, à l’aide de la commande `\unfoldRepeats`, d’une répétition qui commence sur une mesure incomplète et contient un bloc `\alternative` avec modification de la propriété `measureLength` entraînera des messages d’erreur concernant le placement des barres de mesure.

Des reprises imbriquées telles que

```
\repeat ...
\repeat ...
\alternative
```

présentent une ambiguïté, dans la mesure où l’on ne sait à quelle section `\repeat` attribuer la section `\alternative`. Pour résoudre cette ambiguïté, il convient de toujours insérer la commande `\alternative` à l’intérieur de la section `\repeat`. Il est préférable, dans une telle situation, d’utiliser des accolades pour plus de clarté.

Indications de reprise manuelles

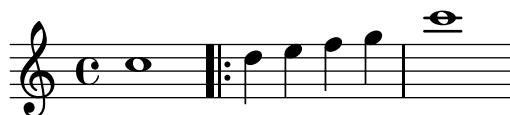
Note : Les méthodes présentées dans les lignes qui suivent ne devraient servir à indiquer que des constructions de répétition inhabituelles. En règle générale, il vaut mieux recourir à la fonction `\repeat` pour créer une reprise ou bien insérer la barre de mesure adéquate. Pour plus d’information, voir le chapitre [Barres de mesure], page 90.

La propriété `repeatCommands` sert à contrôler la mise en forme des reprises. On la définit par une suite de commandes de reprise Scheme.

start-repeat

Pour imprimer une barre de reprise |:

```
c1
\set Score.repeatCommands = #'(start-repeat)
d4 e f g
c1
```



Traditionnellement, on n'imprime pas de signe de reprise en début de morceau.

end-repeat

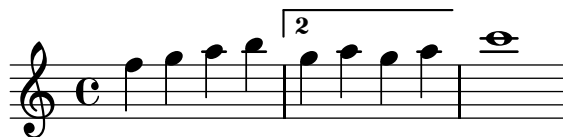
Pour imprimer une barre de reprise :|

```
c1
d4 e f g
\set Score.repeatCommands = #'(end-repeat)
c1
```

**(volta nombre) ... (volta #f)**

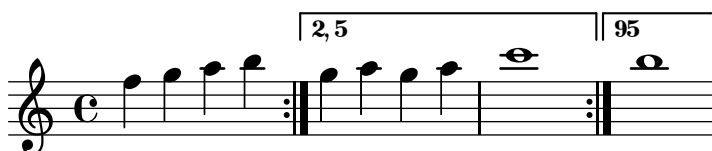
Pour obtenir un crochet indiquant le numéro de l'alternative. Pour que le crochet s'imprime effectivement, il faut spécifier explicitement l'endroit où il doit se terminer.

```
f4 g a b
\set Score.repeatCommands = #'((volta "2"))
g4 a g a
\set Score.repeatCommands = #'((volta #f))
c1
```



Plusieurs commandes de reprise peuvent intervenir au même moment :

```
f4 g a b
\set Score.repeatCommands = #'((volta "2, 5") end-repeat)
g4 a g a
c1
\set Score.repeatCommands = #'((volta #f) (volta "95") end-repeat)
b1
\set Score.repeatCommands = #'((volta #f))
```



Le crochet indiquant une alternative peut contenir aussi du texte. Il peut s'agir d'un ou plusieurs nombres ou bien d'une indication textuelle (*markup*) – voir [Section 1.8.2 \[Mise en forme du texte\]](#), page 221. Le plus simple, dans le cas d'une indication textuelle, est de tout d'abord définir ce *markup*, puis de l'inclure dans une liste Scheme.

```
voltaAdLib = \markup { 1. 2. 3... \text \italic { ad lib. } }
\relative c'' {
  c1
  \set Score.repeatCommands =
    #(list(list 'volta voltaAdLib) 'start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```

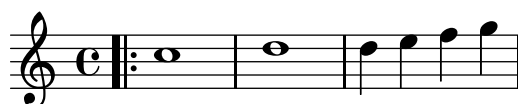


Morceaux choisis

Impression d'une barre de reprise en début de morceau

Bien qu'allant à l'encontre des usages en matière de gravure, une barre de reprise (|:) s'imprimera en début de partition après surcharge de la propriété adéquate :

```
\relative c'' {
  \once \override Score.BreakAlignment #'break-align-orders =
    #(make-vector 3 '(instrument-name
                      left-edge
                      ambitus
                      breathing-sign
                      clef
                      key-signature
                      time-signature
                      staff-bar
                      custos))
  \once \override Staff.TimeSignature #'space-alist =
    #'((first-note . (fixed-space . 2.0))
       (right-edge . (extra-space . 0.5))
       ;; free up some space between time signature
       ;; and repeat bar line
       (staff-bar . (extra-space . 1)))
  \bar "|:"
  c1
  d1
  d4 e f g
}
```



Voir aussi

Manuel de notation : [Barres de mesure], page 90, Section 1.8.2 [Mise en forme du texte], page 221.

Morceaux choisis : Section “Répétitions” dans *Morceaux choisis*.

Référence des propriétés internes : Section “VoltaBracket” dans *Référence des propriétés internes*, Section “RepeatedMusic” dans *Référence des propriétés internes*, Section “VoltaRepeatedMusic” dans *Référence des propriétés internes*.

Répétitions explicites

Adjoindre à la commande `\repeat` l’option `unfold` permet de s’affranchir de ressaisir de la musique répétitive. En voici la syntaxe :

```
\repeat unfold nombre_de_fois expression_musicale
```

Le contenu de *expression_musicale* sera donc expansé autant de fois que stipulé par *nombre_de_fois*.

```
\repeat unfold 2 { c4 d e f }
c1
```



Dans certains cas, et tout particulièrement dans un contexte `\relative`, la fonction `\repeat unfold` ne revient pas à écrire littéralement la même expression musicale plusieurs fois. Ainsi :

```
\repeat unfold 2 { a'4 b c }
```

n’est pas équivalent à

```
a'4 b c | a'4 b c
```

Une répétition expansée peut aussi avoir une fin alternative :

```
\repeat unfold 2 { g4 f e d }
\alternative {
  { c2 g' }
  { cis,2 b }
}
c1
```



Si l’on donne trop peu d’alternatives en regard du nombre de fois où le passage doit être joué, la première alternative sera jouée plusieurs fois.

```
\repeat unfold 4 { c4 d e f }
\alternative {
  { c2 g' }
  { c,2 b }
  { e2 d }
}
c1
```



S'il y a par contre plus d'alternatives que de répétitions, les alternatives superflues seront tout simplement ignorées et ne seront pas imprimées.

```
\repeat unfold 2 { c4 d e f }
\alternative {
  { c2 g' }
  { c,2 b }
  { e2 d }
}
c1
```



Vous pouvez imbriquer plusieurs fonctions `unfold`, qu'elles comportent ou non des fins alternatives :

```
\repeat unfold 2 {
  \repeat unfold 2 { c4 d e f }
  \alternative {
    { c2 g' }
    { c,2 b }
  }
}
c1
```



Une construction en accord peut se répéter à l'aide du symbole `q` – voir [Répétition d'accords], page 151.

Note : L'insertion d'un `\relative` dans une section `\repeat` sans déclaration explicite du contexte `Voice` générera une portée supplémentaire – voir Section “Apparition d'une portée supplémentaire” dans *Utilisation des programmes*.

Voir aussi

Manuel de notation : [Répétition d'accords], page 151.

Morceaux choisis : Section “Répétitions” dans *Morceaux choisis*.

Référence des propriétés internes : Section “RepeatedMusic” dans *Référence des propriétés internes*, Section “UnfoldedRepeatedMusic” dans *Référence des propriétés internes*.

1.4.2 Autres types de répétition

Nous abordons ici les reprises de courte durée. Il en existe deux formes, à savoir la répétition d'une même note sur quelques mesures – représentée par une barre oblique ou le signe pourcent – et les trémolos.

Répétitions de mesure

Le style de « reprise en pourcent » sert à répéter une séquence de notes. Elle sera imprimée une fois, puis remplacée par un symbole spécial.

En voici la syntaxe :

```
\repeat percent nombre expression_musicale
```

Les séquences inférieures à une mesure sont remplacées par une barre oblique.

```
\repeat percent 4 { c128 d e f }
\repeat percent 4 { c64 d e f }
\repeat percent 5 { c32 d e f }
\repeat percent 4 { c16 d e f }
\repeat percent 4 { c8 d }
\repeat percent 4 { c4 }
\repeat percent 2 { c2 }
```



Les séquences d'une ou deux mesures sont remplacées par un symbole qui ressemble au symbole de pourcentage.

```
\repeat percent 2 { c4 d e f }
\repeat percent 2 { c2 d }
\repeat percent 2 { c1 }
```

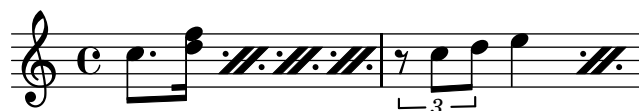


```
\repeat percent 3 { c4 d e f | c2 g' }
```



Les séquences inférieures à la mesure et qui contiennent des durées différentes sont remplacées par un double symbole de pourcentage.

```
\repeat percent 4 { c8. <d f>16 }
\repeat percent 2 { \times 2/3 { r8 c d } e4 }
```



Morceaux choisis

Compteur de répétition en pourcent

Les répétitions de plus de deux mesures sont surmontées d'un compteur, si l'on active la propriété `countPercentRepeats` comme le montre l'exemple suivant :

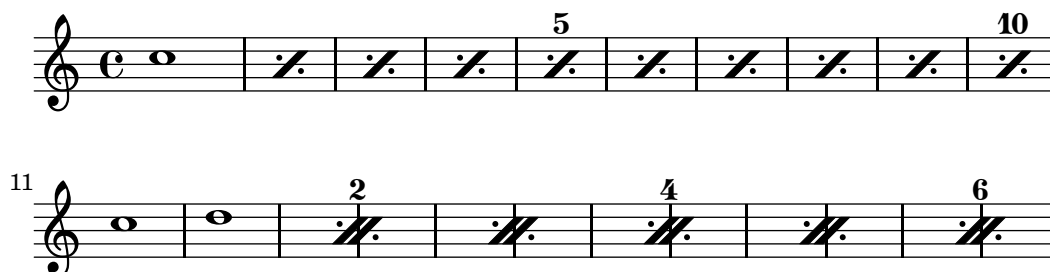
```
\relative c'' {
  \set countPercentRepeats = ##t
  \repeat percent 4 { c1 }
}
```



Affichage du numéro de répétition en pourcent

Le numéro de mesure répétée sera imprimé à intervalle régulier si vous déterminez la propriété de contexte `repeatCountVisibility`.

```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```



Répétition en pourcent isolée

Des symboles de pourcentage isolés peuvent aussi être obtenus, au moyen d'un silence multimesure dont on modifie l'aspect :

```
makePercent =
#(define-music-function (parser location note) (ly:music?)
  "Make a percent repeat the same length as NOTE."
  (make-music 'PercentEvent
    'length (ly:music-length note)))

\relative c'' {
  \makePercent s1
}
```



Voir aussi

Glossaire musicologique : Section “percent repeat” dans *Glossaire*, Section “simile” dans *Glossaire*.

Morceaux choisis : Section “Répétitions” dans *Morceaux choisis*.

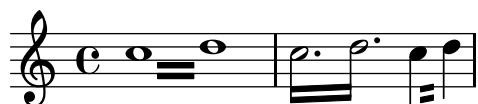
Référence des propriétés internes : Section “RepeatSlash” dans *Référence des propriétés internes*, Section “RepeatSlashEvent” dans *Référence des propriétés internes*, Section “DoubleRepeatSlash” dans *Référence des propriétés internes*, Section “PercentRepeat” dans *Référence des propriétés internes*, Section “PercentRepeatCounter” dans *Référence des propriétés internes*, Section “PercentRepeatedMusic” dans *Référence des propriétés internes*, Section “Percent_repeat_engraver” dans *Référence des propriétés internes*, Section “DoublePercentEvent” dans *Référence des propriétés internes*, Section “DoublePercentRepeat” dans *Référence des propriétés internes*, Section “DoublePercentRepeatCounter” dans *Référence des propriétés internes*, Section “PercentRepeatCounter” dans *Référence des propriétés internes*, Section “PercentRepeatedMusic” dans *Référence des propriétés internes*. Section “Double_percent_repeat_engraver” dans *Référence des propriétés internes*, Section “Slash_repeat_engraver” dans *Référence des propriétés internes*.

Répétitions en trémolo

Il y a deux formes de trémolo : la répétition alternative de deux notes ou accords, et la répétition rapide d’une seule note ou d’un accord. Lorsqu’il est constitué d’une alternance répétitive, le trémolo s’indique en ajoutant des barres de ligature entre les deux notes ou accords concernés. Lorsqu’il s’agit d’une répétition rapide, des barres de ligature penchées sont ajoutées à la note en question.

On peut placer une notation de trémolo entre deux notes, avec la commande `\repeat tremolo` suivie du style trémolo :

```
\repeat tremolo 8 { c16 d }
\repeat tremolo 6 { c16 d }
\repeat tremolo 2 { c16 d }
```



La syntaxe de `\repeat tremolo` requiert expressément deux notes encadrées par des accolades, et le nombre de répétitions exprimé en durée d’une note (pointée ou non). Ainsi, dans l’exemple ci-dessus, `\repeat tremolo 7` est valide car correspond à une note doublement pointée, à l’inverse de `\repeat tremolo 9`.

La durée d’un trémolo est égale à la durée de l’expression entre accolades multipliée par le nombre de fois à répéter : `\repeat tremolo 8 { c16 d16 }` correspond donc à la valeur d’une ronde, et sera représenté par deux rondes séparées par des barres de trémolo.

On peut indiquer de la même manière un trémolo sur une seule note, qu’il faudra alors laisser sans accolades :

```
\repeat tremolo 4 c'16
```



Le même résultat s’obtient en faisant suivre la note considérée de deux points et d’un nombre (`note:nombre`). Le nombre en question correspond à la valeur de la subdivision ; il doit être au

moins de 8, auquel cas la hampe sera barrée par un seul trait de ligature. Si ce nombre est omis, la dernière valeur – telle que mémorisée dans `tremoloFlags` – sera utilisée.

```
c2:8 c:32
c: c:
```



Morceaux choisis

Trémolo et changement de portée

Dans la mesure où `\repeat tremolo` requiert deux arguments musicaux pour un trémolo d'accords, la note ou l'accord de la portée opposée doit être encadré par des accolades et se voir adjoindre la commande `\change Staff`.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
}
>>
```



Voir aussi

Morceaux choisis : [Section “Répétitions”](#) dans *Morceaux choisis*.

1.5 Notes simultanées

The musical score is written for piano in 9/16 time. It consists of three systems of music. The first system begins with a forte (*f*) dynamic and a trill (*tr*) in the right hand. The second system starts at measure 112 with a piano (*p*) dynamic and a trill (*tr*) in the right hand. The third system starts at measure 116 with a piano (*p*) dynamic and a trill (*tr*) in the right hand, followed by a forte (*f*) dynamic change. The score includes various musical notations such as notes, rests, and trills.

La notion musicale de polyphonie fait référence au fait d’avoir plus d’une voix simultanément dans une pièce. Dans LilyPond, la notion de polyphonie fait référence au fait d’avoir plus d’une voix sur la même portée.

1.5.1 Monophonie

Nous allons voir ici comment gérer plusieurs notes simultanées dans une même voix.

Notes en accords

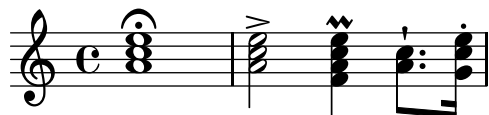
Un accord est formé en mettant une série de hauteurs entre < et >. Un accord peut être suivi d’une durée comme une simple note.

<a c e>1 <a c e>2 <f a c e>4 <a c>8. <g c e>16

The musical notation shows four chords in C major: a triad (C-E-G), a dyad (C-E), a triad (C-E-G), and a dyad (C-E). The chords are written in a single voice on a single staff.

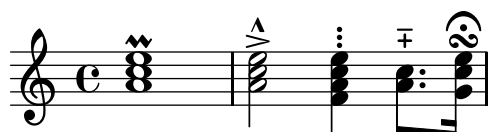
Un accord peut être suivi d'une indication d'articulation comme une simple note.

```
<a c e>1\fermata <a c e>2-> <f a c e>4\prall <a c>8.^| <g c e>16-.
```



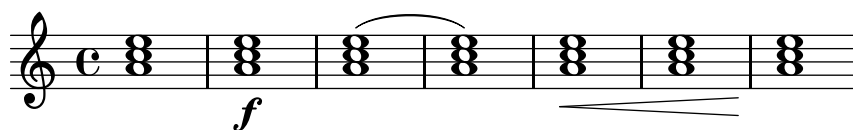
Certaines notes, au sein même d'un accord, peuvent être affectées d'une articulation ou d'un ornement :

```
<a c\prall e>1 <a-> c-^ e>2 <f-. a c-. e-.>4  
<a+ c-->8. <g\fermata c e\turn>16
```



Certains éléments de notation, tels que nuances, soufflets et liaisons, ne peuvent se rattacher qu'à l'accord et non aux notes qui le composent, sous peine de ne les voir s'imprimer.

```
<a\f c( e>1 <a c) e>\f <a\< c e>( <a\! c e>)  
<a c e>\< <a c e> <a c e>\!
```



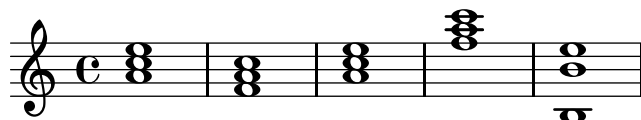
Un accord peut se voir comme un conteneur de notes, articulations et autres éléments rattachés. Par voie de conséquence, un accord dépourvu de note n'a pas de durée ; toute articulation qui lui serait attachée interviendra au même moment que la note ou l'accord qui le suit et lui sera donc combiné – pour des combinaisons plus complexes, voir [Expressions simultanées](#), page 153.

```
\grace { g8[( a b] }  
<> ) \p \< -. -\markup \italic "sempre staccato"  
\repeat unfold 4 { c4 e } c1\f
```



Les accords peuvent être saisis en mode relatif. Dans un accord, l'octave de chaque hauteur saisie est fonction de la précédente, à l'exception de la première qui, elle, sera positionnée en fonction de la première hauteur mentionnée dans l'accord précédent ou de la note individuelle précédente. Les autres notes au sein de l'accord se positionnent relativement à la précédente.

```
<a c e>1 <f a c> <a c e> <f' a c> <b, e b,>
```



Pour plus d'information à propos des accords, voir [Section 2.7 \[Notation des accords\]](#), page 385.

Voir aussi

Glossaire musicologique : [Section “accord”](#) dans *Glossaire*.

Manuel d’initiation : [Section “Combinaison de notes en accords”](#) dans *Manuel d’initiation*.

Manuel de notation : [\[Articulations et ornements\]](#), page 109, [Section 2.7 \[Notation des accords\]](#), page 385. [\[Octaves relatives\]](#), page 2, [Section 1.5.2 \[Plusieurs voix\]](#), page 154.

Morceaux choisis : [Section “Notes simultanées”](#) dans *Morceaux choisis*.

Problèmes connus et avertissements

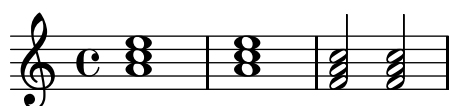
Un accord comportant plus de deux notes dans le même « espace de portée » – tel que ‘<e f! fis!>’ – conduit inmanquablement à des chevauchements. En fonction de la situation, un meilleur rendu peut nécessiter de recourir à

- l’utilisation temporaire de [Section 1.5.2 \[Plusieurs voix\]](#), page 154, ‘<< f! \\ <e fis!> >>’,
- une transcription enharmonique d’une ou plusieurs hauteurs, ‘<e f ges>’, ou
- des [\[Clusters\]](#), page 154.

Répétition d’accords

Dans le but de vous épargner de la saisie, LilyPond dispose d’un raccourci – symbolisé par la lettre q – qui a pour effet de répéter le dernier accord saisi :

```
<a c e>1 q <f a c>2 q
```



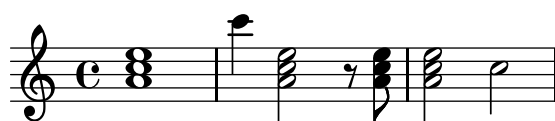
À l’instar de n’importe quel accord, le symbole de répétition peut être affublé d’une durée, de signes d’articulation, *markups*, liaisons, ligatures. . . En fait, c’est la structure du dernier accord qui est dupliquée.

```
<a c e>1\p^"text" q2\<( q8)[-| q8.]!\! q16-1-2-3 q8\prall
```



Dans la mesure où le symbole de répétition d’accord enregistre la structure du dernier accord construit, il est tout à fait possible de l’utiliser même après une succession de notes individuelles et de silences :

```
<a c e>1 c'4 q2 r8 q8 |
q2 c, |
```



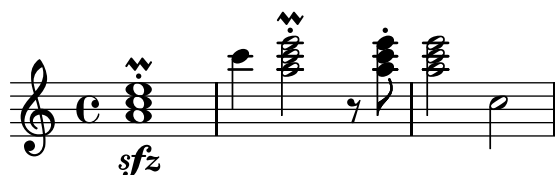
Le symbole de répétition d’accord ne prend en charge que les hauteurs, en aucun cas les nuances, articulations ou ornements, qu’elles aient été attachées aux notes le composant ou à l’ensemble.

```
<a-. c\prall e>1\sfz c4 q2 r8 q8 |
q2 c, |
```



Le seul moyen de les reproduire consiste à utiliser explicitement la fonction `\chordRepeats`, en lui adjoignant un argument supplémentaire qui recense les *types d'événement* à répéter et qui seraient absents de l'accord construit par un `q`.

```
\relative c'' {
  \chordRepeats #'(articulation-event)
  { <a-. c\prall e>1\sfz c'4 q2 r8 q8-. } |
  q2 c, |
}
```



Comme vous pouvez le constater, l'utilisation de `\chordRepeats` au sein d'un bloc `\relative` ne produit pas le résultat escompté : les événements de l'accord expansés sont identiques à la saisie traditionnelle d'un accord, ce qui a pour conséquence que l'octave affectée par `\relative` repose sur le contexte en cours.

Dans la mesure où l'imbrication de clauses `\relative` n'est pas source d'interférence, l'ajout d'un `\relative` à ce qui sera concerné par l'instruction `\chordRepeats` permet d'établir une relation d'octave entre les accords dès avant leur expansion. Dans le cas présent, l'intégralité du bloc `\relative` intérieur n'affecte en rien ce qui l'entoure, ce qui explique la spécification d'octave attachée à la dernière note.

```
\new Voice
\relative c'' {
  \chordRepeats #'(articulation-event)
  \relative c''
  { <a-. c\prall e>1\sfz c'4 q2 r8 q8-. } |
  q2 c |
}
```



Les interactions avec `\relative` ne se produisent que lors d'un appel explicite de `\chordRepeats` : l'expansion implicite en début de saisie intervient à un moment où toutes les instances de `\relative` ont déjà été interprétées.

Voir aussi

Manuel de notation : [\[Articulations et ornements\]](#), page 109, [Section 2.7 \[Notation des accords\]](#), page 385.

Fichiers d'initialisation : `'ly/chord-repetition-init.ly'`.

Expressions simultanées

Lorsqu'une ou plusieurs expressions musicales sont encadrées par des doubles chevrons, elles sont considérées comme étant simultanées. Si la première expression débute par une note unique ou si l'intégralité de l'expression simultanée est explicitement rattachée à une voix en particulier, elle sera placée sur une seule portée. Dans le cas contraire, les éléments d'une expression simultanée seront placés sur des portées distinctes.

Voici deux exemples d'expression simultanée sur une même portée :

```
\new Voice { % explicit single voice
  << { a4 b g2 } { d4 g c,2 } >>
}
```



```
% single first note
a << { a4 b g } { d4 g c, } >>
```



Cette manière de procéder est utile lorsque les éléments de l'expression ont des rythmes identiques. Dès que vous tenterez d'attacher sur une même hampe des notes de durée différente, vous générerez des erreurs. Notes, articulations et modifications de propriétés au sein d'un même *Voice* sont enregistrés et gravés selon l'ordre musical :

```
<a c>4-. <>-. << c a >> << { c-. <c a> } { a s-. } >>
```



La présence de plusieurs différentes hampes, ligatures, durées ou propriétés au même instant musical nécessite l'utilisation de plusieurs voix.

Dans l'exemple suivant, l'expression simultanée génère implicitement plusieurs portées :

```
% no single first note
<< { a4 b g2 } { d4 g2 c,4 } >>
```



En pareil cas, des rythmes différents ne sont source d'aucun problème puisqu'ils sont interprétés dans des voix différentes.

Problèmes connus et avertissements

Des notes, bien qu'appartenant à des voix différentes, mais dont les hampes ont la même orientation, peuvent se retrouver au même endroit sur la portée, ce quelque soit le décalage que vous auriez pu leur appliquer. Ceci ne manque pas de faire apparaître un message

```
warning: ignoring too many clashing note columns
```

en français :

Avertissement : trop d'empilements de notes se chevauchent. On fera au mieux.■

lors de la compilation. Le déclenchement de cet avertissement peut être désactivé par une clause

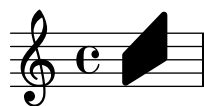
```
\override NoteColumn #'ignore-collision = ##t
```

Ceci n'aura pas pour seule conséquence que ce message ne sera plus émis ; les procédures d'évitement de collision de quelqu'ordre que ce soit seront désactivées, ce qui peut conduire à quelques effets inattendus (voir aussi *Problèmes connus et avertissements* à la rubrique [Résolution des collisions], page 158).

Clusters

Un cluster indique un agrégat de sons. On peut le représenter par une plage limitée par un *ambitus* (notes extrêmes). On obtient une telle notation en appliquant la fonction `makeClusters` à une séquence d'accords, comme

```
\makeClusters { <g b>2 <c g'> }
```



Des notes ordinaires et des clusters peuvent cohabiter sur une même portée, y compris simultanément – en pareil cas, rien ne sera fait pour tenter d'empêcher les chevauchements entre notes et clusters.

Voir aussi

Glossaire musicologique : Section “cluster” dans *Glossaire*.

Morceaux choisis : Section “Notation simultanée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “ClusterSpanner” dans *Référence des propriétés internes*, Section “ClusterSpannerBeacon” dans *Référence des propriétés internes*, Section “cluster-spanner-engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

L'apparence d'un cluster sera extrêmement resserrée s'il ne comporte au moins deux accords.

Dans la mesure où un cluster ne possède pas de hampe, il n'y a aucun moyen d'en connaître la durée ; cependant la longueur du signe imprimé dépend directement de la durée affectée aux accords qui le définissent.

Seul un silence peut séparer deux clusters.

Les clusters ne sont pas reproduits en MIDI.

1.5.2 Plusieurs voix

Nous allons nous intéresser, dans les paragraphes qui suivent, à la gestion de notes simultanées réparties sur plusieurs voix ou plusieurs portées.

Polyphonie sur une portée

Instanciation explicite des voix

La manière la plus facile d'entrer des fragments avec plus d'une voix sur une portée est la suivante :

```
\new Staff <<
  \new Voice = "first"
    { \voiceOne r8 r16 g e8. f16 g8[ c,] f e16 d }
  \new Voice= "second"
    { \voiceTwo d16 c d8~ d16 b c8~ c16 b c8~ c16 b8. }
>>
```



Vous constaterez que les voix sont créées explicitement et qu'elles sont nommées. Les commandes `\voiceOne ... \voiceFour` déterminent les voix de telle sorte que les première et troisième auront des hampes vers le haut, et les deuxième et quatrième vers le bas. D'autre part, les notes des troisième et quatrième voix seront quelque peu décalées, tout comme leurs silences, afin d'éviter les collisions. La commande `\oneVoice` permet de retrouver les réglages par défaut.

Polyphonie temporaire

Un fragment temporairement polyphonique se construit de la manière suivante :

```
<< { \voiceOne ... }
  \new Voice { \voiceTwo ... }
>> \oneVoice
```

En fait, la première expression d'une polyphonie temporaire reste dans le même contexte `Voice` que celui existant auparavant et qui perdurera après ce fragment. Les autres expressions entre doubles chevrons seront assignées à des voix temporaires distinctes. C'est la raison pour laquelle les paroles qui suivaient la voix avant la polyphonie continueront à le faire durant ce passage polyphonique et après lui :

```
<<
  \new Voice = "melody" {
    a4
    <<
      {
        \voiceOne
        g f
      }
      \new Voice {
        \voiceTwo
        d2
      }
    >>
    \oneVoice
    e4
  }
  \new Lyrics \lyricsto "melody" {
    This is my song.
  }
>>
```



Vous remarquerez que les commandes `\voiceOne` et `\voiceTwo` permettent d'obtenir des réglages différents pour chacune des voix.

La construction avec un double antislash

Une construction de la forme `<< { ... } \\ { ... } >>`, dans laquelle plusieurs expressions sont séparées par des doubles obliques inversées, se comporte différemment de celle sans séparateur : **tous** les membres de cette construction seront assignés à de nouveaux contextes de voix. Ces contextes de voix, créés implicitement, portent les noms "1", "2", etc. Dans chacun de ces contextes, le positionnement des liaisons, la direction des hampes, etc. sont réglés de manière appropriée. En voici un exemple :

```
<<
  { r8 r16 g e8. f16 g8[ c,] f e16 d }
  \\
  { d16 c d8~ d16 b c8~ c16 b c8~ c16 b8. }
>>
```



Cette syntaxe peut être utilisée dans la mesure où la création puis la disparition de voix temporaires sont sans conséquence. Les réglages de ces voix créées implicitement sont les mêmes que si elles avaient été créées à l'aide des commandes `\voiceOne` à `\voiceFour`, dans leur ordre d'apparition.

Dans l'exemple qui suit, la voix intermédiaire a des hampes vers le haut. Nous la plaçons donc en troisième position, de telle sorte qu'elle adopte les réglages de `\voiceThree` qui correspondent à ce que nous voulons. Grâce à des espaces invisibles, nous évitons de surcharger la portée avec des demis soupirs.

```
<<
  { r8 g g g g f16 ees f8 d }
  \\
  { ees,8 r ees r d r d r }
  \\
  { d'8 s c s bes s a s }
>>
```



En dehors des cas les plus simples, nous vous invitons à toujours créer les contextes de voix de manière explicite. Voir à ce sujet [Section "Contextes et graveurs" dans Manuel d'initiation](#) et [Section "Instanciation explicite des voix" dans Manuel d'initiation](#).

Ordre des voix

L'ordre dans lequel doivent apparaître les voix d'une construction simultanée suit le schéma suivant :

Voix 1: la plus haute
 Voix 2: la plus basse
 Voix 3: deuxième plus haute
 Voix 4: deuxième plus basse
 Voix 5: troisième plus haute
 Voix 6: troisième plus basse
 etc.

Cette présentation en entonnoir peut sembler quelque peu contre-intuitive ; elle simplifie cependant grandement le processus de mise en forme. Vous noterez que les hampes des voix au numéro impair vont vers le haut, celles des voix paires vers le bas :

```
\new Staff <<
  \time 2/4
  { f''2 } % 1: highest
  \\
  { c'2 } % 2: lowest
  \\
  { d''2 } % 3: second-highest
  \\
  { e'2 } % 4: second-lowest
  \\
  { b'2 } % 5: third-highest
  \\
  { g'2 } % 6: third-lowest
>>
```



Note : Paroles et objets étendus (liaisons, soufflets etc.) ne peuvent passer d'une voix à l'autre.

Identité rythmique

Lorsque l'on doit saisir des fragments de musique parallèle qui ont le même rythme, on peut les combiner dans un contexte de voix unique et par voie de conséquence former des accords. Il suffit pour cela de les regrouper dans une construction de musique simultanée simple au sein d'une voix explicite :

```
\new Voice <<
  { e4 f8 d e16 f g8 d4 }
  { c4 d8 b c16 d e8 b4 }
>>
```



Prenez garde que les différents éléments doivent impérativement avoir la même structure rythmique, sous peine de ligature aléatoire et de messages d'avertissement.

Commandes prédéfinies

`\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`, `\oneVoice`.

Voir aussi

Manuel d’initiation : Section “Instanciation explicite des voix” dans *Manuel d’initiation*, Section “Les voix contiennent la musique” dans *Manuel d’initiation*.

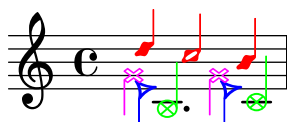
Manuel de notation : [Hampes], page 207, [Portées de percussion], page 365, [Silences invisibles], page 53.

Morceaux choisis : Section “Notation simultanée” dans *Morceaux choisis*.

Styles de voix

Opter pour des couleurs et des têtes de notes spécifiques selon la voix permet de les identifier plus facilement :

```
<<
{ \voiceOneStyle d4 c2 b4 }
\\
{ \voiceTwoStyle e,2 e }
\\
{ \voiceThreeStyle b2. c4 }
\\
{ \voiceFourStyle g'2 g }
>>
```



La commande `\voiceNeutralStyle` permet de revenir à la présentation normale.

Commandes prédéfinies

`\voiceOneStyle`, `\voiceTwoStyle`, `\voiceThreeStyle`, `\voiceFourStyle`, `\voiceNeutralStyle`.

Voir aussi

Manuel d’initiation : Section “Autres sources de documentation” dans *Manuel d’initiation*, Section “J’entends des Voix” dans *Manuel d’initiation*.

Morceaux choisis : Section “Notation simultanée” dans *Morceaux choisis*.

Résolution des collisions

Les notes de hauteur identique appartenant à des voix différentes, même si leur hampe sont opposées, verront leur tête automatiquement fusionner. Les notes dont la tête diffère ou bien qui ont la hampe dans la même direction ne seront pas automatiquement fusionnées. Les silences, lorsqu’ils sont dans une autre voix et à l’opposé des hampes seront décalés verticalement. Vous constaterez, dans l’exemple suivant, que la fusion échoue aux premier et troisième temps de la première mesure, ainsi qu’au premier temps de la deuxième mesure.

```
<<
{
c8 d e d c d c4
g'2 fis
```



```

} \ {
  c2 c8. b16 c4
  e,2 r
} \ {
  \oneVoice
  s1
  e8 a b c d2
}
>>

```



Cependant, vous pouvez fusionner une tête de blanche avec une tête de croche – jamais avec une noire. Les têtes du premier temps de la première mesure ont bien fusionné :

```

<<
{
  \mergeDifferentlyHeadedOn
  c8 d e d c d c4
  g'2 fis
} \ {
  c2 c8. b16 c4
  e,2 r
} \ {
  \oneVoice
  s1
  e8 a b c d2
}
>>

```



De même, vous pouvez fusionner les têtes de notes pointées et non pointées comme au troisième temps de la première mesure :

```

<<
{
  \mergeDifferentlyHeadedOn
  \mergeDifferentlyDottedOn
  c8 d e d c d c4
  g'2 fis
} \ {
  c2 c8. b16 c4
  e,2 r
} \ {
  \oneVoice
  s1
  e8 a b c d2
}
>>

```

```
}
>>
```



Lorsque trois notes ou plus s'agglutinent dans un même empilement, `\mergeDifferentlyHeadedOn` ne peut mener à bien la fusion des deux notes qui devraient l'être. Pour obtenir une fusion optimale, appliquez un décalage (`\shift`) à la note qui ne devrait pas fusionner. Ici, on applique un `\shiftOn` pour décaler le *sol* de l'empilement ; le rendement de `\mergeDifferentlyHeadedOn` est alors comme il faut.

```
<<
{
  \mergeDifferentlyHeadedOn
  \mergeDifferentlyDottedOn
  c8 d e d c d c4
  \shiftOn
  g'2 fis
} \ {
  c2 c8. b16 c4
  e,2 r
} \ {
  \oneVoice
  s1
  e8 a b c d2
}
>>
```



La commande `\shiftOn` permet, sans pour autant le forcer, un décalage des notes d'une voix en particulier. Une note ou un accord appartenant à cette voix ne seront décalés que si leur hampe menaçait d'entrer en collision avec une hampe appartenant à une autre voix allant dans la même direction. La commande `\shiftOff` interdit l'apparition de décalage.

Les voix externes – habituellement les voix une et deux – sont affectées de `\shiftOff`, alors que les voix internes – trois et quatre – sont affectées de `\shiftOn`. Lorsqu'un décalage intervient, les notes dont les hampes sont ascendantes (voix impaire) iront vers la droite, et les notes à hampe descendante (voix paire) iront vers la gauche.

Voici un exemple qui vous permettra de bien visualiser ce qui se passe en interne.

Note : Lorsqu'il y a trois voix ou plus, prenez garde au fait que l'ordre d'apparition des voix dans votre fichier ne correspond pas à l'ordre vertical des voix tel qu'il apparaîtra sur la portée.

```
\new Staff \relative c'' {
  %% saisie abrégée
  <<
```

```

    { f2 } % 1: extrême haute
    \\\
    { g,2 } % 2: extrême basse
    \\\
    { d'2 } % 3: intermédiaire haute
    \\\
    { b2 } % 4: intermédiaire basse
  >>
  %% expansion en interne de ce qui précède
  <<
    \new Voice = "1" { \voiceOne \shiftOff f'2 }
    \new Voice = "2" { \voiceTwo \shiftOff g,2 }
    \new Voice = "3" { \voiceThree \shiftOn d'2 } % décale à droite
    \new Voice = "4" { \voiceFour \shiftOn b2 } % décale à gauche
  >>
}

```



Les commandes `\shiftOnn` et `\shiftOnnn` sont des niveaux supplémentaires de décalage qui peuvent s'adopter temporairement dans certaines situations complexes – voir [Section “Exemple concret”](#) dans *Manuel d'initiation*.

Les têtes de notes ne fusionneront que dans la mesure où leur hampe sont opposées – implicitement parce qu'appartenant aux voix une ou deux, ou bien explicitement.

Commandes prédéfinies

`\mergeDifferentlyDottedOn`, `\mergeDifferentlyDottedOff`, `\mergeDifferentlyHeadedOn`, `\mergeDifferentlyHeadedOff`.

`\shiftOn`, `\shiftOnn`, `\shiftOnnn`, `\shiftOff`.

Morceaux choisis

Ajout de voix pour éviter les collisions

Dans certains cas de musique polyphonique complexe, une voix supplémentaire peut permettre d'éviter les risques de collision. Lorsque quatre voix parallèles ne suffisent pas, la fonction `Scheme context-spec-music` permet d'ajouter encore d'autres voix.

```

voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)

\relative c'' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
  >>
}

```

```

\\
{
  \voiceThree
  f,2
  bes4 a2
  a4 s2
}
\\
{
  \voiceFive
  s2
  g4 g2
  f4 f2
}
\\
\bar "||" {
  \voiceTwo
  d2
  d4 cis2
  d4 bes2
}
>>
}

```



Décalage horizontal forcé

Quand LilyPond est dépassé, la propriété `force-hshift` de l'objet `NoteColumn`, et des silences à hauteur déterminée, peuvent s'avérer utiles pour dicter au programme les choix de placement. On travaille ici en espace de portée.

```

\relative c' <<
{
  <d g>2 <d g>
}
\\
{
  <b f'>2
  \once \override NoteColumn #'force-hshift = #1.7
  <b f'>2
}
>>

```



Voir aussi

Glossaire musicologique : [Section “polyphonie” dans *Glossaire*](#).

Manuel d’initiation : [Section “Exemple concret” dans *Manuel d’initiation*](#), [Section “Les voix contiennent la musique” dans *Manuel d’initiation*](#), [Section “Notes simultanées” dans *Manuel d’initiation*](#).

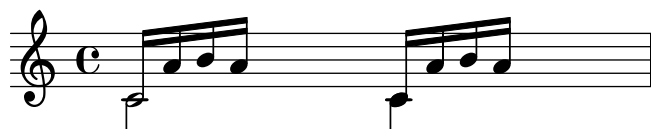
Morceaux choisis : [Section “Notation simultanée” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “NoteColumn” dans *Référence des propriétés internes*](#), [Section “NoteCollision” dans *Référence des propriétés internes*](#), [Section “RestCollision” dans *Référence des propriétés internes*](#).

Problèmes connus et avertissements

Une clause `\override NoteColumn #'ignore-collision = ##t` aura pour effet une fusion incorrecte des têtes de note différentes à partir de voix différentes.

```
\mergeDifferentlyHeadedOn
<< { c16 a' b a } \ { c,2 } >>
\override NoteColumn #'ignore-collision = ##t
<< { c16 a' b a } \ { c,2 } >>
```



Regroupement automatique de parties

Le regroupement automatique de parties vous permet de fusionner deux pupitres sur une seule portée, ceci dans le but de créer des partitions d’orchestre. Lorsque les deux parties sont identiques sur une certaine durée, une seule s’affiche. Lorsqu’elles diffèrent, deux voix séparées apparaissent, avec des hampes dont la direction est gérée automatiquement. Vous pouvez aussi identifier et faire ressortir les solos et parties *a due*.

Voici la syntaxe qui permet de combiner des parties :

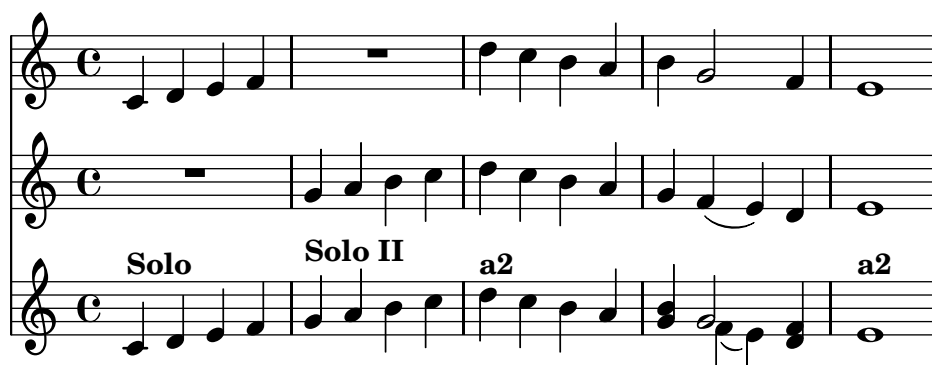
```
\partcombine expression_musicale_1 expression_musicale_2
```

L’exemple suivant illustre les fonctionnalités élémentaires du combineur de parties : positionner les parties sur une portée, gérer la direction des hampes et de la polyphonie. Les identifiants sont les mêmes pour la combinaison et les parties séparées.

```
instrumentOne = \relative c' {
  c4 d e f |
  R1 |
  d'4 c b a |
  b4 g2 f4 |
  e1 |
}

instrumentTwo = \relative g' {
  R1 |
  g4 a b c |
  d4 c b a |
  g4 f( e) d |
  e1 |
}
```

```
<<
  \new Staff \instrumentOne
  \new Staff \instrumentTwo
  \new Staff \partcombine \instrumentOne \instrumentTwo
>>
```



Les notes de la troisième mesure n'apparaissent qu'une seule fois, alors qu'elles ont été spécifiées deux fois (une fois dans chacune des parties). La direction des hampes et des liaisons de tenue ou de phrasé est gérée automatiquement, selon qu'il s'agisse d'un solo ou d'un unisson. La première partie, dont le contexte s'appellera **one**, aura toujours ses hampes dirigées vers le haut et sera notée « Solo », alors que la deuxième, appelée **two**, aura des hampes vers le bas et sera notée « Solo II ». Les parties à l'unisson seront estampillées d'un « a2 » par défaut.

LilyPond interprète dans un contexte **Voice** chacun des arguments fournis à **\partcombine**. Si vous travaillez avec des octaves relatives, spécifiez **\relative** dans chacune des expressions musicales, comme ceci :

```
\partcombine
  \relative ... expression_musicale_1
  \relative ... expression_musicale_2
```

Une section **\relative** à l'extérieur du **\partcombine** restera sans effet sur les hauteurs de **expression_musicale_1** ou de **expression_musicale_2**.

En matière d'édition professionnelle, les voix sont souvent maintenues séparément et sur une durée conséquente, bien que les notes des différentes voix soient les mêmes et pourraient donc être présentées à l'unisson. Dans la mesure où **\partcombine** considère les notes séparément, combiner des notes en accord ou indiquer un solo ne serait pas optimal en pareil cas. LilyPond dispose alors de certaines commandes qui permettent d'influencer le comportement de la fonction **\partcombine** :

Les commandes se terminant par **...Once** n'affectent que la note qui les suit directement dans l'expression musicale.

- **\partcombineApart** et **\partcombineApartOnce** maintiennent les notes dans des voix séparées même si elles peuvent se combiner en accord ou en unisson.
- **\partcombineChords** et **\partcombineChordsOnce** combinent les notes en accords.
- **\partcombineUnisono** et **\partcombineUnisonoOnce** combinent les voix en un « unisson ».
- **\partcombineSoloI** et **\partcombineSoloIOnce** affichent exclusivement la première voix et l'affublent d'un « Solo ».
- **\partcombineSoloII** et **\partcombineSoloIIOnce** affichent exclusivement la deuxième voix et l'affublent d'un « Solo ».

- `\partcombineAutomatic` et `\partcombineAutomaticOnce` annulent les effets des dérogations précédentes et activent le comportement standard de la fonction `\partcombine`.

```

instrumentOne = \relative c' {
  \partcombineApart c2^"apart" e |
  \partcombineAutomatic e2^"auto" e |
  \partcombineChords e'2^"chord" e |
  \partcombineAutomatic c2^"auto" c |
  \partcombineApart c2^"apart" \partcombineChordsOnce e^"chord once" |
  c2 c |
}
instrumentTwo = \relative c' {
  c2 c |
  e2 e |
  a,2 c |
  c2 c' |
  c2 c |
  c2 c |
}

<<
  \new Staff { \instrumentOne }
  \new Staff { \instrumentTwo }
  \new Staff { \partcombine \instrumentOne \instrumentTwo }
>>

```

The image displays a musical score for three staves. The top staff is labeled with 'apart', 'auto', 'chord', 'auto', 'apart', and 'chord once' above the notes. The middle staff is labeled with 'apart', 'a2', 'chord', 'auto', 'a2', and 'apart' above the notes. The bottom staff is labeled with 'apart', 'a2', 'chord', 'auto', 'a2', and 'apart' above the notes. The notes are written in a standard musical notation with a treble clef and a common time signature (C).

Morceaux choisis

Combinaison de deux parties sur une même portée

L'outil de combinaison de parties (la commande `\partcombine`) permet d'avoir deux parties différentes sur une même portée. LilyPond ajoute automatiquement des indications textuelles, telles que « solo » ou « a2 ». Si votre intention n'est que de fusionner les parties, sans ajouter de texte, assignez faux à la propriété `printPartCombineTexts`. Dans le cas de partitions vocales, et plus particulièrement d'hymnes, ces « solo/a2 » ne sont d'aucune utilité, aussi vaut-il mieux les désactiver. Dans le cas où il y aurait alternance entre *solo* et *tutti*, il vaut mieux faire appel à de la musique polyphonique standard.

Voici trois moyens d'imprimer deux parties sur une même portée : en polyphonie normale, avec `\partcombine` sans indication supplémentaire, et avec `\partcombine` commentée.

```

musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}

\score {
  <<
    <<
      \new Staff {
        \set Staff.instrumentName = #"Standard polyphony"
        << \musicUp \\\musicDown >>
      }
      \new Staff \with { printPartCombineTexts = ##f } {
        \set Staff.instrumentName = #"PartCombine without texts"
        \partcombine \musicUp \musicDown
      }
      \new Staff {
        \set Staff.instrumentName = #"PartCombine with texts"
        \partcombine \musicUp \musicDown
      }
    >>
  >>
  \layout {
    indent = 6.0\cm
    \context {
      \Score
      \override SystemStartBar #'collapse-height = #30
    }
  }
}

```

Standard polyphony



PartCombine without texts



PartCombine with texts





Modification des indications de parties combinées

Lorsque vous regroupez automatiquement des parties, vous pouvez modifier le texte qui sera affiché pour les solos et pour les parties à l'unisson :

```
\new Staff <<
  \set Staff.soloText = #"girl"
  \set Staff.soloIIText = #"boy"
  \set Staff.aDueText = #"together"
  \partcombine
    \relative c'' {
      g4 g r r
      a2 g
    }
    \relative c'' {
      r4 r a( b)
      a2 g
    }
  >>
```



Voir aussi

Glossaire musicologique : Section “a due” dans *Glossaire*, Section “partie” dans *Glossaire*.

Manuel de notation : Section 1.6.3 [Écriture de parties séparées], page 188.

Morceaux choisis : Section “Notation simultanée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “PartCombineMusic” dans *Référence des propriétés internes*, Section “Voice” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les différentes commandes `\partcombine...` ne prennent en charge que deux voix. De la même manière, le combinateur n’est pas conçu pour travailler avec des paroles ; il s’arrête dès qu’il est explicitement fait appel à l’une des voix pour y attacher des paroles.

`\partcombine...` ne peut s’inscrire ni dans un bloc `\times`, ni dans un bloc `\relative`.

Lorsque `printPartCombineTexts` est actif et que les deux voix jouent souvent les mêmes notes, le combinateur peut afficher `a2` plus d’une fois par mesure.

`\partcombine` n’examine que l’attaque des notes. Il n’est donc pas en mesure de déterminer si une note attaquée précédemment est encore jouée ou non, ce qui peut engendrer quelques problèmes.

En interne, `\partcombine` interprète les deux arguments en tant que *Voices*, dénommées *one* et *two*, puis décide de quand les parties seront fusionnées. Par conséquent, si les arguments changent pour d'autres noms de contexte *Section “Voice” dans Référence des propriétés internes*, les événements qu'ils contiendraient seront ignorés.

Certaines considérations apparaissent aussi dans les chapitres [Tablatures par défaut], page 319 et [Barres de ligature automatiques], page 75.

Saisie de musique en parallèle

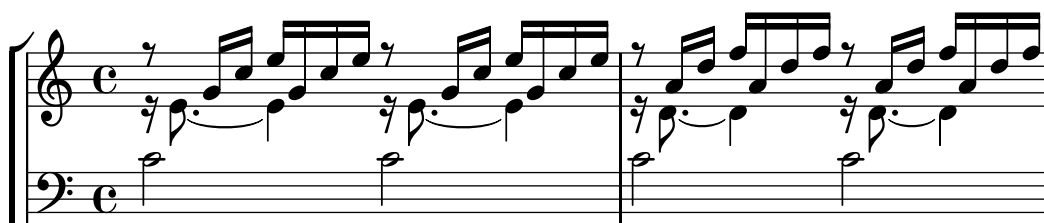
On peut écrire plusieurs voix de façon entremêlée. La fonction `\parallelMusic` prend en charge une liste des variables à créer, ainsi qu'une expression musicale. Le contenu des différentes mesures de l'expression musicale deviennent les valeurs des variables respectives que vous pourrez ensuite utiliser pour imprimer la partition.

Note : Les contrôles de barre de mesure `|` sont obligatoires et les mesures doivent être de longueur identique.

```
\parallelMusic #'(voiceA voiceB voiceC) {
  % Bar 1
  r8 g'16 c'' e'' g' c'' e'' r8 g'16 c'' e'' g' c'' e'' |
  r16 e'8.~ e'4 r16 e'8.~ e'4 |
  c'2 c'2 |

  % Bar 2
  r8 a'16 d'' f'' a' d'' f'' r8 a'16 d'' f'' a' d'' f'' |
  r16 d'8.~ d'4 r16 d'8.~ d'4 |
  c'2 c'2 |

}
\new StaffGroup <<
  \new Staff << \voiceA \ \voiceB >>
  \new Staff { \clef bass \voiceC }
>>
```



Vous pouvez travailler en mode relatif. Notez cependant que la commande `\relative` n'apparaît pas au sein du bloc `\parallelMusic`. Le calcul des hauteurs relatives s'effectue voix par voix, et non au fil des lignes saisies ; en d'autres termes, les notes de la *voiceA* ignorent tout de celles de la *voiceB*.

```
\parallelMusic #'(voiceA voiceB voiceC) {
  % Bar 1
  r8 g16 c e g, c e r8 g,16 c e g, c e |
  r16 e8.~ e4 r16 e8.~ e4 |
  c2 c |

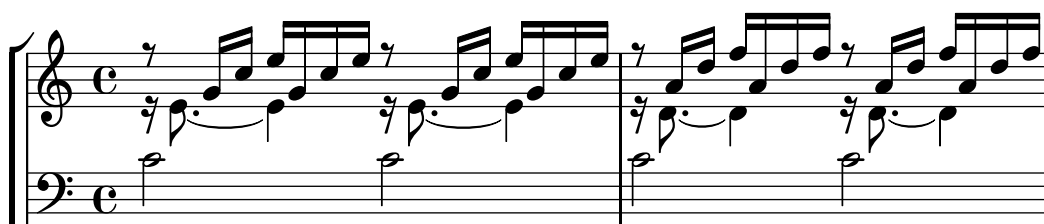
  % Bar 2
```

```

r8 a,16 d f a, d f r8 a,16 d f a, d f |
r16 d8.~ d4          r16 d8.~ d4          |
c2                  c                  |

}
\new StaffGroup <<
  \new Staff << \relative c'' \voiceA \\ \relative c' \voiceB >>
  \new Staff \relative c' { \clef bass \voiceC }
>>

```



Ceci fonctionne bien avec la musique pour piano. L'exemple suivant affecte quatre mesures à quatre variables :

```

global = {
  \key g \major
  \time 2/4
}

\parallelMusic #'(voiceA voiceB voiceC voiceD) {
  % Bar 1
  a8    b    c    d    |
  d4          e    |
  c16 d e fis d e fis g |
  a4          a    |

  % Bar 2
  e8    fis g    a    |
  fis4    g    |
  e16 fis g a fis g a b |
  a4          a    |

  % Bar 3 ...
}

\score {
  \new PianoStaff <<
    \new Staff {
      \global
      <<
        \relative c'' \voiceA
        \\
        \relative c' \voiceB
      >>
    }
    \new Staff {
      \global \clef bass
    }
  >>
}

```

```

    <<
      \relative c \voiceC
      \\\
      \relative c \voiceD
    >>
  }
>>
}

```



Voir aussi

Manuel d'initiation : [Section "Organisation du code source avec des variables"](#) dans *Manuel d'initiation*.

Morceaux choisis : [Section "Notation simultanée"](#) dans *Morceaux choisis*.

1.6 Notation sur la portée

Cette section aborde les détails de gravure de la portée, la réalisation de partitions comprenant plusieurs portées et l'ajout d'indications globales d'exécution, présentes sur certaines portées seulement.

1.6.1 Gravure des portées

Nous allons voir ici comment créer des portées et comment les regrouper.

Initialisation de nouvelles portées

Les *portées* – en anglais *staff* (*staves* au pluriel) – sont créées à l’aide des commandes `\new` ou `\context`. Pour de plus amples détails, consultez [Section 5.1.2 \[Création d’un contexte\]](#), page 551.

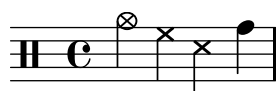
Le contexte de portée standard s’appelle `Staff` :

```
\new Staff { c4 d e f }
```



Le contexte `DrumStaff` crée une portée à cinq lignes correspondant à une batterie traditionnelle et chacun des instruments est représenté par un symbole spécifique. Les éléments sont saisis en mode batterie, initialisé par la commande `\drummode`, chaque composante étant spécifiée par son nom. Pour de plus amples détails, consultez [\[Portées de percussion\]](#), page 365.

```
\new DrumStaff {
  \drummode { cymc hh ss tomh }
}
```



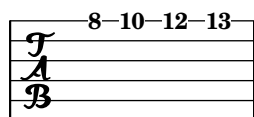
Un `RhythmicStaff` est composé d’une portée à ligne unique chargée de présenter les valeurs rythmiques saisies. Seules sont imprimées les durées. Pour de plus amples détails, consultez [\[Gravure de lignes rythmiques\]](#), page 73.

```
\new RhythmicStaff { c4 d e f }
```



Un `TabStaff` crée une portée de tablature correspondant aux six cordes d’une guitare standard. Pour de plus amples détails, consultez [\[Tablatures par défaut\]](#), page 319.

```
\new TabStaff { c4 d e f }
```



LilyPond dispose aussi de deux contextes dédiés à la musique ancienne : `MensuralStaff` et `VaticanaStaff`. Ils sont abordés plus en détails au chapitre [\[Contextes prédéfinis\]](#), page 409.

Le contexte `GregorianTranscriptionStaff` permet d’obtenir des éditions modernes du grégorien. Bien entendu, il est dépourvu de barres de mesure.

```
\new GregorianTranscriptionStaff { c4 d e f e d }
```



Vous pourrez toujours définir d’autres contextes de portée selon vos besoins, en suivant les indications fournies au chapitre [Section 5.1.6 \[Définition de nouveaux contextes\]](#), page 562.

Voir aussi

Glossaire musicologique : Section “staff” dans *Glossaire*, Section “portées” dans *Glossaire*.

Manuel de notation : Section 5.1.2 [Création d’un contexte], page 551, [Portées de percussion], page 365, [Gravure de lignes rythmiques], page 73, [Tablatures par défaut], page 319, [Contextes prédéfinis], page 409, [Symbole de la portée], page 178, [Contextes du chant grégorien], page 419, [Contextes de musique mensurale], page 411, Section 5.1.6 [Définition de nouveaux contextes], page 562.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

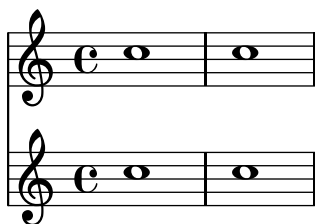
Référence des propriétés internes : Section “Staff” dans *Référence des propriétés internes*, Section “DrumStaff” dans *Référence des propriétés internes*, Section “GregorianTranscriptionStaff” dans *Référence des propriétés internes*, Section “RhythmicStaff” dans *Référence des propriétés internes*, Section “TabStaff” dans *Référence des propriétés internes*, Section “MensuralStaff” dans *Référence des propriétés internes*, Section “VaticanaStaff” dans *Référence des propriétés internes*, Section “StaffSymbol” dans *Référence des propriétés internes*.

Regroupement de portées

LilyPond dispose de différents contextes permettant de regrouper des portées individuelles et d’obtenir ainsi des « systèmes ». Chacun de ces contextes définira le style de regroupement, avec son signe particulier en début de portée et ses règles de gestion des barres de mesure.

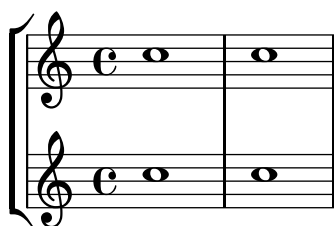
Lorsqu’aucun contexte particulier n’est spécifié, les propriétés suivantes s’appliqueront par défaut : les portées du groupe ne sont pas reliées, hormis par une simple ligne verticale en début de ligne, et les barres de mesure sont indépendantes.

```
<<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



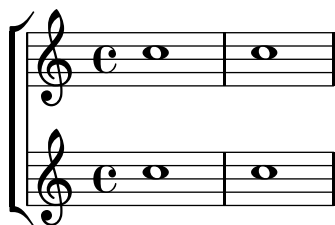
Dans un `StaffGroup`, le groupe de portées est signifié par un crochet, et les barres de mesure sont d’un seul tenant.

```
\new StaffGroup <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



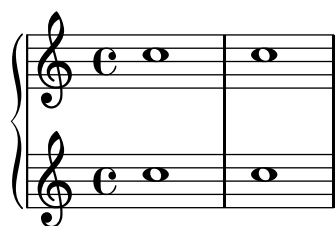
Dans un `ChoirStaff`, le groupe de portées est signifié par un crochet sur la gauche, et les barres de mesure sont individuelles.

```
\new ChoirStaff <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



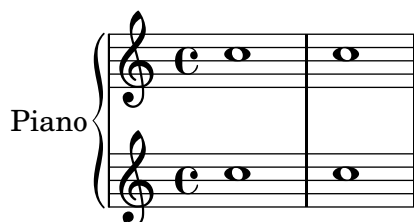
Dans un `GrandStaff`, le groupe de portées est signifié par une accolade sur la gauche, et les barres de mesure sont d'un seul tenant.

```
\new GrandStaff <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



Le contexte `PianoStaff` est identique au `GrandStaff`, sauf qu'il gère automatiquement l'affichage du nom d'instrument – voir [\[Noms d'instrument\]](#), page 188 pour plus de détails.

```
\new PianoStaff <<
  \set PianoStaff.instrumentName = #"Piano"
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



Les barres de mesure au début de chaque système adoptent l'un des styles `SystemStartBar`, `SystemStartBrace` ou `SystemStartBracket`. Dans chaque contexte, seul l'un de ces styles est utilisé, et c'est la propriété `systemStartDelimiter` qui détermine lequel. Un quatrième style, `SystemStartSquare`, doit quant à lui être spécifié de manière explicite.

Vous pouvez aussi créer vos propres contextes de regroupement, comme l'explique [Section 5.1.6 \[Définition de nouveaux contextes\]](#), page 562.

Morceaux choisis

Indication de regroupement de portées par un rectangle

Un regroupement de portées sera indiqué par un simple rectangle – `SystemStartSquare` – en début de ligne dès lors que vous le mentionnerez explicitement au sein d'un contexte `StaffGroup` ou `ChoirStaff`.

```
\score {
  \new StaffGroup { <<
    \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



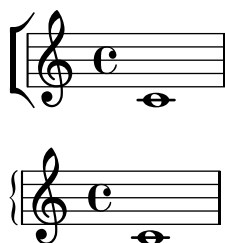
Indicateur de regroupement et portée unique

Lorsque, dans des regroupements de type `ChoirStaff` ou `StaffGroup`, une seule portée est active, aucune indication n'est donnée en début de ligne. Surcharger la propriété adéquate permet de modifier ce comportement par défaut.

Notez bien que dans le cas des `PianoStaff` et `GrandStaff`, pour lesquels le délimiteur de système est une accolade et non un crochet, il ne s'agit pas de la même propriété – voir le deuxième système de l'exemple.

```
\markup \left-column {
  \score {
    \new StaffGroup <<
      % Must be lower than the actual number of staff lines
      \override StaffGroup.SystemStartBracket #'collapse-height = #1
      \override Score.SystemStartBar #'collapse-height = #1
      \new Staff {
        c'1
      }
    >>
    \layout { }
  }
  \null
  \score {
    \new PianoStaff <<
      \override PianoStaff.SystemStartBrace #'collapse-height = #1
      \override Score.SystemStartBar #'collapse-height = #1
      \new Staff {
        c'1
      }
    >>
    \layout { }
  }
}
```

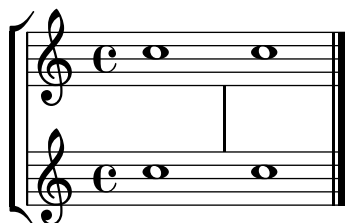

}



Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat avec un `StaffGroup` plutôt qu'en utilisant un `ChoirStaff`, il faudra rendre « transparentes » les portions de barre qui recouvrent les portées.

```
global = {
  \override Staff.BarLine #'transparent = ##t
  s1 s
  % the final bar line is not interrupted
  \revert Staff.BarLine #'transparent
  \bar "|."
}
\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```



Voir aussi

Glossaire musicologique : Section “accolade” dans *Glossaire*, Section “crochet” dans *Glossaire*, Section “système” dans *Glossaire*.

Manuel de notation : [Noms d’instrument], page 188, Section 5.1.6 [Définition de nouveaux contextes], page 562.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Staff” dans *Référence des propriétés internes*, Section “StaffGroup” dans *Référence des propriétés internes*, Section “ChoirStaff” dans *Référence des propriétés internes*, Section “GrandStaff” dans *Référence des propriétés internes*, Section “PianoStaff” dans *Référence des propriétés internes*, Section “SystemStartBar” dans *Référence des propriétés internes*, Section “SystemStartBrace” dans *Référence des propriétés internes*, Section “SystemStartBracket” dans *Référence des propriétés internes*, Section “SystemStartSquare” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Un `PianoStaff` ne peut contenir, par défaut, de `ChordNames`.

Imbrication de regroupements de portées

Les accolades et crochets qui délimitent les systèmes peuvent être imbriqués en profondeur. Chaque niveau inférieur aura son propre délimiteur, en plus de celui du niveau supérieur.

```
\new StaffGroup <<
  \new Staff { c2 c | c2 c }
  \new StaffGroup <<
    \new Staff { g2 g | g2 g }
    \new StaffGroup \with {
      systemStartDelimiter = #'SystemStartSquare
    }
    <<
      \new Staff { e2 e | e2 e }
      \new Staff { c2 c | c2 c }
    >>
  >>
>>
```



Vous pouvez aussi créer vos propres contextes d'imbrication, comme l'explique [Section 5.1.6 \[Définition de nouveaux contextes\]](#), page 562.

Morceaux choisis

Imbrications de regroupements de portées

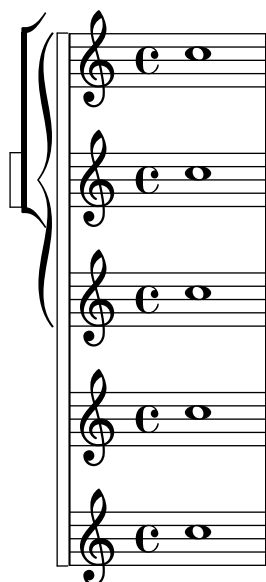
La propriété `systemStartDelimiterHierarchy` permet de créer des regroupements imbriqués complexes. La commande `\set StaffGroup.systemStartDelimiterHierarchy` prend en argument la liste alphabétique des sous-groupes à hiérarchiser. Chaque sous-groupe peut être affublé d'un délimiteur particulier. Chacun des regroupements intermédiaires doit être borné par des parenthèses. Bien que des éléments de la liste puissent être omis, le premier délimiteur embrassera toujours l'intégralité des portées. Vous disposez des quatre délimiteurs `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` et `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBrace (SystemStartBracket a
      (SystemStartSquare b) ) c ) d)
  \new Staff { c1 }
  \new Staff { c1 }
```

```

\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
>>

```



Voir aussi

Manuel de notation : [Regroupement de portées], page 172, [Noms d’instrument], page 188, Section 5.1.6 [Définition de nouveaux contextes], page 562.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “StaffGroup” dans *Référence des propriétés internes*, Section “ChoirStaff” dans *Référence des propriétés internes*, Section “SystemStartBar” dans *Référence des propriétés internes*, Section “SystemStartBrace” dans *Référence des propriétés internes*, Section “SystemStartBracket” dans *Référence des propriétés internes*, Section “SystemStartSquare” dans *Référence des propriétés internes*.

Séparation des systèmes

Le nombre de systèmes peut varier d’une page à l’autre ; vous pouvez, en pareil cas, rendre plus évidente la séparation entre les systèmes en l’indiquant visuellement. Ce « séparateur » est absent par défaut, mais vous pouvez l’activer par une option au sein du bloc `\paper`.

```

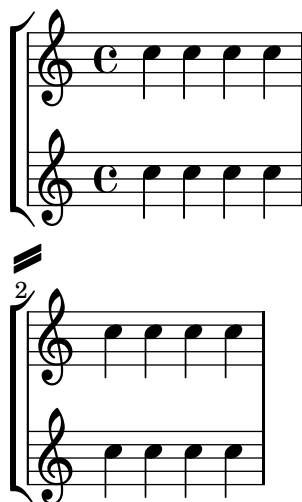
\book {
  \score {
    \new StaffGroup <<
      \new Staff {
        \relative c'' {
          c4 c c c
          \break
          c4 c c c
        }
      }
    }
  \new Staff {
    \relative c'' {
      c4 c c c
      \break
    }
  }
}

```

```

        c4 c c c
      }
    }
  >>
}
\paper {
  system-separator-markup = \slashSeparator
  % following commands are needed only to format this documentation
  paper-width = 100\mm
  paper-height = 100\mm
  tagline = ##f
}
}

```



Voir aussi

Manuel de notation : [Section 4.1 \[Mise en forme de la page\]](#), page 496.

Morceaux choisis : [Section “Notation sur la portée”](#) dans *Morceaux choisis*.

1.6.2 Modification de portées individuelles

Cette section explique le réglage de la gravure de chaque portée, comme la taille de portée ou le nombre de lignes ; sont aussi décrits la suspension et la reprise de portées, ainsi que les portées d'*ossia*.

Symbole de la portée

Les commandes `\stopStaff` et `\startStaff` permettent respectivement de clôturer et (re)démarrer une portée n'importe où dans une partition.

```

\stopStaff f4 d \startStaff g, e
f'4 d \stopStaff g, e
f'4 d \startStaff g, e

```



Commandes prédéfinies

`\startStaff`, `\stopStaff`.

Les notes, nuances, etc. sont regroupées dans un assemblage de lignes horizontales, que l'on nomme la portée (en anglais *staff*, et *staves* au pluriel). Dans LilyPond, ces lignes sont dessinées au moyen d'un objet graphique (*grob*) à part entière, nommé `StaffSymbol` – symbole de portée. Modifier les propriétés d'un `StaffSymbol` changera l'apparence de la portée, dès lors qu'elles auront été définies avant de créer la portée en question.

Vous pouvez modifier le nombre de lignes d'une portée.

```
f4 d \stopStaff
\override Staff.StaffSymbol #'line-count = #2
\startStaff g, e |

f'4 d \stopStaff
\revert Staff.StaffSymbol #'line-count
\startStaff g, e |
```



Le positionnement de chacune des lignes de la portée est modifiable. Les valeurs s'expriment en *demi* espace de portée, et la nouvelle position est déterminée par rapport à la ligne médiane. Une ligne sera donc imprimée pour chaque valeur exprimée ; le nombre de lignes, ainsi que leur position dans la portée, peut donc se modifier à l'aide d'une seule commande.

```
f4 d \stopStaff
\override Staff.StaffSymbol #'line-positions = #'(1 3 5 -1 -3)
\startStaff g, e |
f'4 d \stopStaff
\override Staff.StaffSymbol #'line-positions = #'(8 6.5 -6 -8 -0.5)
\startStaff g, e
```



La position de la clef et celle du do médium demanderont parfois un ajustement afin d'être en phase avec cette nouvelle portée. Pour plus d'explications, reportez-vous aux exemples du chapitre [\[Clefs\]](#), page 16.

Lorsque vous modifierez l'épaisseur des lignes, gardez à l'esprit que les lignes supplémentaires et les hampes seront aussi modifiées.

```
\new Staff \with {
  \override StaffSymbol #'thickness = #3
}
{ e4 d c b }
```



L'épaisseur des lignes supplémentaires (*ledger lines*) peut être déterminée indépendamment des lignes de la portée. Dans l'exemple suivant, les deux nombres constituent des coefficients

multiplicateurs de l'épaisseur de la ligne. Ils sont fournis en argument pour déterminer l'épaisseur des lignes supplémentaires.

```
\new Staff \with {
  \override StaffSymbol #'thickness = #2
  \override StaffSymbol #'ledger-line-thickness = #'(0.5 . 0.4)
}
{ f'4 a, a,, f }
```



L'emplacement des lignes supplémentaires est réglable :

```
\new Staff \with {
  \override StaffSymbol #'ledger-positions = #'(-3 -2 -1 2 5 6)
}
{ f'4 a, a,, f }
```



Vous pouvez faire apparaître des lignes supplémentaires additionnelles au-dessus ou en dessous des têtes de note selon leur positionnement relatif aux autres notes, qu'elles aient ou non elles-mêmes des lignes supplémentaires.

```
\new Staff \with {
  \override StaffSymbol #'ledger-extra = #4
}
{ f'4 a, d, f, }
```

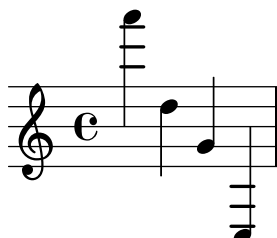


Des lignes supplémentaires peuvent apparaître y compris au sein d'une portée, notamment lorsque vous l'avez personnalisée. L'exemple suivant illustre deux cas de figure quant au positionnement des lignes supplémentaires selon que la propriété `ledger-position` est définie explicitement ou non. La présence du `stopStaff` est ici rendue nécessaire pour annuler les effets de la commande `\override` qui s'applique à l'intégralité du `StaffSymbol`.



Modifier l'équidistance des lignes de la portée affectera aussi les lignes supplémentaires.

```
\new Staff \with {
  \override StaffSymbol #'staff-space = #1.5
}
{ f'4 d, g, e, }
```



Morceaux choisis

Empâtement de certaines lignes d'une portée

Vous pourriez avoir envie, dans un but pédagogique, de rendre certaines lignes d'une portée plus épaisses que les autres, comme la ligne médiane, ou bien pour mettre en exergue la ligne portant la clef de sol. Il suffit pour cela d'ajouter une ligne qui sera accolée à celle qui doit être mise en évidence, grâce à la propriété `line-positions` de l'objet `StaffSymbol`.

```
{
  \override Staff.StaffSymbol #'line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Voir aussi

Glossaire musicologique : [Section “ligne” dans *Glossaire*](#), [Section “ligne supplémentaire” dans *Glossaire*](#), [Section “portée” dans *Glossaire*](#).

Manuel de notation : [\[Clefs\]](#), page 16.

Morceaux choisis : [Section “Notation sur la portée” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “StaffSymbol” dans *Référence des propriétés internes*](#), [Section “staff-symbol-interface” dans *Référence des propriétés internes*](#).

Portées d'ossia

Une portée d'ossia – ou de variante – s'obtient en créant, à l'endroit approprié, une nouvelle portée simultanée :

```
\new Staff \relative c'' {
  c4 b d c
  <<
    { c4 b d c }
    \new Staff { e4 d f e }
  >>
  c4 b c2
}
```



L'exemple ci-dessus n'est probablement pas ce qui vous conviendra le plus. Afin que cette ossia se place au-dessus de la portée à laquelle elle se réfère, étant par ailleurs dépourvue de métrique et de clef, et d'une taille légèrement inférieure, vous devrez avoir recours à quelques retouches. Le manuel d'initiation aborde une technique particulière pour obtenir ce résultat, au chapitre [Section "Expressions musicales imbriquées"](#) dans *Manuel d'initiation*.

L'exemple qui suit utilise, pour aligner la portée d'ossia, la propriété `alignAboveContext`. Cette méthode est tout à fait appropriée lorsqu'il y a un nombre restreint d'ossias.

```
\new Staff = main \relative c' ' {
  c4 b d c
  <<
    { c4 b d c }

    \new Staff \with {
      \remove "Time_signature_engraver"
      alignAboveContext = #"main"
      fontSize = #-3
      \override StaffSymbol #'staff-space = #(magstep -3)
      \override StaffSymbol #'thickness = #(magstep -3)
      firstClef = ##f
    }
    { e4 d f e }
  >>
  c4 b c2
}
```



Dans le cas où de nombreux et courts fragments d'ossia affecteraient une même portée, il est judicieux de créer un contexte `Staff` vide auquel sera attribué un *identifiant*. Il suffira alors, pour entamer un fragment d'ossia, de « faire appel » à ce contexte grâce aux commandes `\startStaff` et `\stopStaff`. Vous vous rendrez compte à l'utilisation des avantages que procure cette façon de procéder, bien plus que dans l'exemple suivant.

```
<<
  \new Staff = ossia \with {
    \remove "Time_signature_engraver"
    \override Clef #'transparent = ##t
    fontSize = #-3
    \override StaffSymbol #'staff-space = #(magstep -3)
    \override StaffSymbol #'thickness = #(magstep -3)
  }
  { \stopStaff s1*6 }
```



```

\new Staff \relative c' {
  c4 b c2
  <<
    { e4 f e2 }
    \context Staff = ossia {
      \startStaff e4 g8 f e2 \stopStaff
    }
  >>
  g4 a g2 \break
  c4 b c2
  <<
    { g4 a g2 }
    \context Staff = ossia {
      \startStaff g4 e8 f g2 \stopStaff
    }
  >>
  e4 d c2
}
>>

```



4



Vous pourriez aussi recourir à la commande `\Staff \RemoveEmptyStaves` pour créer votre portée d'ossia. Cependant, cette méthode reste limitée à l'apparition de ces ossias en début de ligne. Pour plus d'information au sujet de la commande `\Staff \RemoveEmptyStaves`, reportez-vous au chapitre [\[Masquage de portées\]](#), page 185.

```

<<
\new Staff = ossia \with {
  \remove "Time_signature_engraver"
  \override Clef #'transparent = ##t
  fontSize = #-3
  \override StaffSymbol #'staff-space = #(magstep -3)
  \override StaffSymbol #'thickness = #(magstep -3)
} \relative c'' {
  R1*3
  c4 e8 d c2
}
\new Staff \relative c' {
  c4 b c2
  e4 f e2
}

```

```

g4 a g2 \break
c4 b c2
g4 a g2
e4 d c2
}
>>

\layout {
  \context {
    \Staff \RemoveEmptyStaves
    \override VerticalAxisGroup #'remove-first = ##t
  }
}

```



Morceaux choisis

Positionnement d'une ossia et des paroles

Cet exemple illustre la manière de positionner une portée d'ossia et des paroles à l'aide des propriétés de contexte `alignBelowContext` et `alignAboveContext`.

```

\paper {
  ragged-right = ##t
}

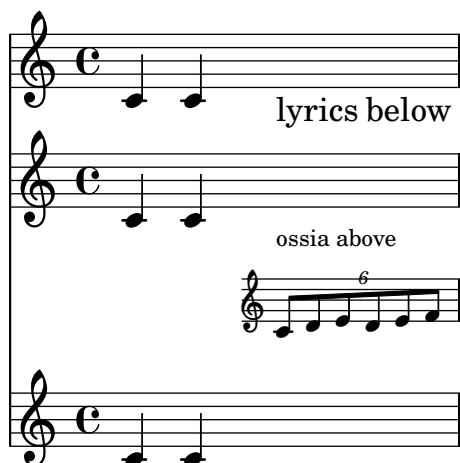
\relative c' <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = #"3"
        fontSize = #-2
        \override StaffSymbol #'staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
      } {
        \times 4/6 {

```

```

\override TextScript #'padding = #3
c8[~"ossia above" d e d e f]
}
}
>>
}
>>

```



Voir aussi

Glossaire musicologique : Section “*ossia*” dans *Glossaire*, Section “*portée*” dans *Glossaire*, Section “*Frenched staff*” dans *Glossaire*.

Manuel d'initiation : Section “Expressions musicales imbriquées” dans *Manuel d'initiation*, Section “Taille des objets” dans *Manuel d'initiation*, Section “Longueur et épaisseur des objets” dans *Manuel d'initiation*.

Manuel de notation : [Masquage de portées], page 185.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “StaffSymbol” dans *Référence des propriétés internes*.

Masquage de portées

Désactiver le graveur `Staff_symbol_engraver` dans un contexte `Staff` permet de masquer des lignes. La commande `\stopStaff` aura le même effet.

```
\new Staff \with {
  \remove "Staff_symbol_engraver"
}
\relative c''' { a8 f e16 d c b a2 }
```

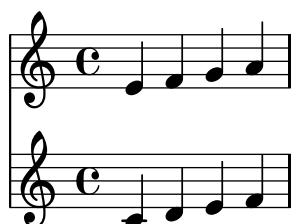


L’instruction `\Staff \RemoveEmptyStaves` placée dans un bloc `\layout` aura pour effet de masquer toute portée qui ne contient rien. Dans les partitions d’orchestre, les portées qui n’ont que des silences sont habituellement masquées afin de gagner de la place. Ce style d’édition s’appelle en anglais « French Score ». Cette fonctionnalité masque et supprime toutes les portées vides d’une partition, hormis celles du premier système.

Note : Une portée est considérée comme vide dès lors qu'elle ne contient que des silences multimesures, des silences visibles ou invisibles (ou d'espacement – les *\skip*) ou bien une combinaison de ces éléments.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
  }
}

\relative c' <<
  \new Staff {
    e4 f g a \break
    b1 \break
    a4 b c2
  }
  \new Staff {
    c,4 d e f \break
    R1 \break
    f4 g c,2
  }
  >>
```



`\Staff \RemoveEmptyStaves` permet aussi de gérer des fragments d'ossia attachés à une portée. Pour plus de détails, voir [\[Portées d'ossia\]](#), page 181.

La commande `\VaticanaStaff \RemoveEmptyStaves` permet de masquer des portées vides dans les contextes de musique ancienne. Pour des contextes `RhythmicStaff`, il faudra utiliser `\RhythmicStaff \RemoveEmptyStaves`.

Commandes prédéfinies

`\Staff \RemoveEmptyStaves`, `\VaticanaStaff \RemoveEmptyStaves`, `\RhythmicStaff \RemoveEmptyStaves`.

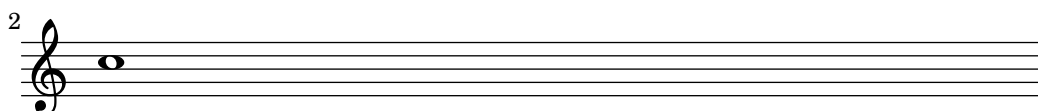
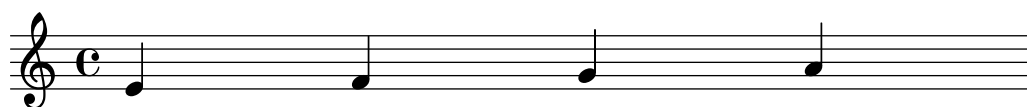
Morceaux choisis

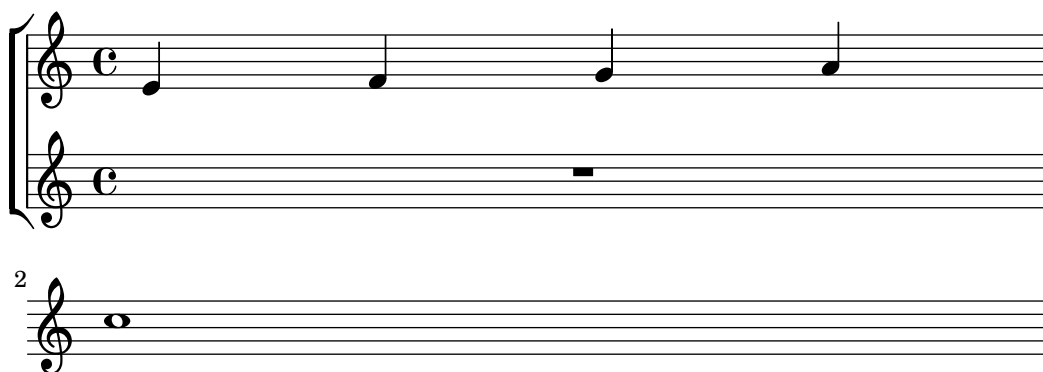
Masquage de la première ligne si elle est vide

Par défaut, le premier système comportera absolument toutes les portées. Si vous préférez masquer les portées vides y compris pour le premier système, vous devrez activer la propriété **remove-first** du **VerticalAxisGroup**. Mentionnée dans un bloc `\layout`, cette commande agira de manière globale. Pour qu'elle ne soit effective que pour une portée particulière, vous devrez également spécifier le contexte (**Staff** pour qu'il ne concerne que la portée en cours) en préfixe de la propriété.

La première ligne inférieure du deuxième **StaffGroup** est bien présente, pour la simple raison que le réglage en question ne s'applique qu'à la portée dans laquelle il a été inscrit.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup #'remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup #'remove-first = ##t
    R1 \break
    R
  }
}
>>
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
}
>>
```





Voir aussi

Glossaire musicologique : Section “Frenched staff” dans *Glossaire*.

Manuel d’initiation: Section “Visibilité et couleur des objets” dans *Manuel d’initiation*.

Manuel de notation : [Symbole de la portée], page 178, [Portées d’ossia], page 181, [Dictée à trous], page 204, [Silences invisibles], page 53, Section 5.4.6 [Visibilité des objets], page 586.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “ChordNames” dans *Référence des propriétés internes*, Section “FiguredBass” dans *Référence des propriétés internes*, Section “Lyrics” dans *Référence des propriétés internes*, Section “Staff” dans *Référence des propriétés internes*, Section “VerticalAxisGroup” dans *Référence des propriétés internes*, Section “Staff_symbol_engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Supprimer le `Staff_symbol_engraver` aura pour effet de masquer les barres de mesure. Forcer leur visibilité peut entraîner des problèmes de formatage. En pareil cas il vaut mieux, au lieu de supprimer le graveur, recourir aux dérogations suivantes :

```
\override StaffSymbol #'stencil = ##f
\override NoteHead #'no-ledgers = ##t
```

En ce qui concerne les désagréments et messages liés à l’utilisation de l’instruction `\Staff \RemoveEmptyStaves`, consultez [Section 5.1.5 \[Modification des réglages par défaut d’un contexte\]](#), page 557.

1.6.3 Écriture de parties séparées

Nous allons voir, au fil des lignes qui suivent, comment insérer des indications de tempo ou des noms d’instrument dans une partition. Nous aborderons aussi la citation d’autres voix, et comment la mettre en forme.

Noms d’instrument

Dans un conducteur, les noms d’instrument sont portés en regard de chacune des portées, qu’il s’agisse d’un contexte `Staff`, `PianoStaff`, `StaffGroup`, `GrandStaff` ou `ChoirStaff`. La première ligne affichera la valeur de `instrumentName`, et les suivantes celle de `shortInstrumentName`.

```
\new Staff \with {
  instrumentName = #"Violin "
  shortInstrumentName = #"Vln. "
}
{ c4.. g'16 c4.. g'16 \break | c1 }
```





Le recours à la commande `\markup` permet de construire des noms d'instrument particuliers, tels que

```
\new Staff \with {
  instrumentName = \markup {
    \column { "Clarinetti"
      \line { "in B" \smaller \flat }
    }
  }
}
{ c4 c,16 d e f g2 }
```



Lorsque plusieurs contextes de portée sont regroupés, les noms d'instrument, que ce soit sous leur forme développée ou abrégée, sont par défaut centrés. Si l'un d'entre eux est libellé sur plusieurs lignes, il faudra recourir à l'instruction `\center-column` :

```
<<
\new Staff \with {
  instrumentName = #"Flute"
}
{ f2 g4 f }
\new Staff \with {
  instrumentName = \markup {
    \center-column { "Clarinet" }
    \line { "in B" \smaller \flat }
  }
}
{ c4 b c2 }
>>
```



Lorsque le nom d'un instrument est relativement long, il est judicieux d'augmenter les retraits – *indent* – au sein du bloc `\layout` à l'aide des commandes `indent` et `short-indent`. Pour plus de plus amples détails sur ces réglages, reportez-vous au chapitre [\[Variables d'indentation et de décalage\]](#), page 503.

```
\relative c'' {
  <<
  \new Staff \with {
    instrumentName = #"Alto Flute in G"
    shortInstrumentName = #"Flt."
  }
  {
```

```

        f2 g4 f \break
        g4 f g2
    }
    \new Staff \with {
        instrumentName = #"Clarinet"
        shortInstrumentName = #"Clar."
    }
    {
        c,4 b c2 \break
        c2 b4 c
    }
    >>
}

\layout {
    indent = 3.0\cm
    short-indent = 1.5\cm
}

```

Alto Flute in G

Clarinet

Flt.

Clar.

Des noms d'instrument peuvent s'utiliser dans d'autres contextes, tels que `ChordNames` ou `FiguredBass`, dès lors qu'on leur adjoint le graveur `Instrument_name_engraver`. Pour de plus amples informations sur la manière d'activer ou désactiver un graveur, voir [Section 5.1.4 \[Modification des greffons de contexte\]](#), page 555.

Vous pouvez changer le nom d'un instrument en cours de morceau. Notez cependant que la valeur de `instrumentName` ne s'affichera que sur la première portée :

```

\new Staff \with {
    instrumentName = #"Flute"
    shortInstrumentName = #"Flt."
}
{
    c1 c c c \break
    c1 c c c \break
    \set Staff.instrumentName = #"Clarinet"
    \set Staff.shortInstrumentName = #"Clt."
    c1 c c c \break
    c1 c c c \break
}

```


}

Lorsqu'un musicien doit changer d'instrument – piccolo et flûte, basson et contre-basson –, la commande `\addInstrumentDefinition`, couplée à l'instruction `\instrumentSwitch` permet de spécifier en détail les modifications intervenant lors du changement. La commande `\addInstrumentDefinition` prend deux arguments : une chaîne de caractères qui servira d'identifiant, et une liste d'association de valeurs aux propriétés de ce nouvel instrument. Ces définitions devront être déclarées avant tout autre élément musical. L'instruction `\instrumentSwitch` se placera dans la musique au moment de la substitution :

```
\addInstrumentDefinition #"contrabassoon"
#`((instrumentTransposition . ,(ly:make-pitch -1 0 0))
  (shortInstrumentName . "Cbsn.")
  (clefGlyph . "clefs.F")
  (middleCPosition . 6)
  (clefPosition . 2)
  (instrumentCueName . ,(make-bold-markup "cbsn.))
  (midiInstrument . "bassoon"))
```

```
\new Staff \with {
  instrumentName = #"Bassoon"
}
\relative c' {
  \clef tenor
  \compressFullBarRests
  c2 g'
  R1*16
  \instrumentSwitch "contrabassoon"
  c,,2 g \break
  c,1 ~ | c1
}
```



Voir aussi

Manuel de notation : [Variables d’indentation et de décalage], page 503, Section 5.1.4 [Modification des greffons de contexte], page 555.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “InstrumentName” dans *Référence des propriétés internes*, Section “PianoStaff” dans *Référence des propriétés internes*, Section “Staff” dans *Référence des propriétés internes*.

Citation d’autres voix

Il est assez courant qu’une voix soit doublée par une autre. Par exemple, les premiers et seconds violons peuvent jouer les mêmes notes durant un moment. LilyPond gère parfaitement ces situations où une voix est la *réplique* d’une autre, sans devoir ressaisir la musique en question.

L’instruction `\addQuote`, placée au niveau le plus haut – c’est à dire en dehors de tout bloc de musique – définit la musique dont il sera possible de répliquer des fragments.

Au cours d’une partie, des extraits de répliques peuvent être cités en utilisant la commande `\quoteDuring`. Cette commande prend deux arguments : le nom de la voix reproduite, tel que défini par `\addQuote`, et une expression musicale qui indique la durée de cette citation.

```
fluteNotes = \relative c'' {
  a4 gis g gis | b4~"quoted" r8 ais\p a4( f)
}

oboeNotes = \relative c'' {
  c4 cis c b \quoteDuring #"flute" { s1 }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```



Si l’expression musicale utilisée pour `\quoteDuring` contenait autre chose que du silence, il en résulterait une situation polyphonique, ce qui n’est pas le but recherché :

```
fluteNotes = \relative c'' {
  a4 gis g gis | b4~"quoted" r8 ais\p a4( f)
}
```

```

oboeNotes = \relative c'' {
  c4 cis c b \quoteDuring #"flute" { e4 r8 ais b4 a }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}

```

L'instruction `\quoteDuring` prendra en compte les réglages d'une commande `\transposition`, qu'elle apparaisse au niveau de la voix répliquée ou dans celle qui réplique.

```

clarinetNotes = \relative c'' {
  \transposition bes
  \key d \major
  b4 ais a ais | cis4^"quoted" r8 bis\p b4( f)
}

oboeNotes = \relative c'' {
  c4 cis c b \quoteDuring #"clarinet" { s1 }
}

\addQuote "clarinet" { \clarinetNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Clarinet" } \clarinetNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}

```

LilyPond répliquera, par défaut, tous les éléments – articulations, nuances, *markups*, etc. La propriété `quotedEventTypes` permet de définir plus précisément quels éléments de la voix originelle seront reproduits.

```
fluteNotes = \relative c'' {
  a2 g2 |
  b4\<^"quoted" r8 ais a4\f( c->)
}

oboeNotes = \relative c'' {
  c2. b4 |
  \quoteDuring #"flute" { s1 }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \set Score.quotedEventTypes = #'(note-event articulation-event
                                      crescendo-event rest-event
                                      slur-event dynamic-event)
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```



Les citations peuvent être « balisées » par un nom particulier afin de les utiliser de différentes manières. Pour de plus amples détails à ce propos, consultez le chapitre [\[Utilisation de balises\]](#), page 477.

Voir aussi

Manuel de notation : [\[Instruments transpositeurs\]](#), page 22, [\[Utilisation de balises\]](#), page 477.

Fichiers d’initialisation : ‘`scm/define-event-classes.scm`’.

Morceaux choisis : [Section “Notation sur la portée”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Music classes”](#) dans *Référence des propriétés internes*, [Section “QuoteMusic”](#) dans *Référence des propriétés internes*, [Section “Voice”](#) dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Seul le contenu de la première `Voice` rencontrée dans la partie marquée d’une commande `\addQuote` pourra être retenu. Par voie de conséquence, *musique* ne saurait comprendre de `\new` ou une instance `\context Voice` qui la ferait passer à une autre voix.

Citer des notes d'ornement ne fonctionne pas, et peut même entraîner un blocage de LilyPond.

Citer des triolets imbriqués peut entraîner un résultat de piètre qualité.

Mise en forme d'une citation

Le moyen le plus simple pour mettre en forme des notes provenant d'une autre voix consiste à déclarer explicitement un contexte `CueVoice` au sein de la voix où elle apparaît.

```
R1
<<
{ e2\rest r4. e8 }
\new CueVoice {
  \stemUp d'8^"flute" c d e fis2
}
>>
d,,4 r a r
```



L'instruction `\cueClef`, utilisée conjointement à un contexte `CueVoice` explicite permet d'indiquer la clef, dans une taille réduite, propre à la voix citée. Le retour à la clef d'origine s'effectue à l'aide de l'instruction `\cueClefUnset`.

```
\clef "bass"
R1
<<
{ e2\rest r4. \cueClefUnset e,8 }
\new CueVoice {
  \cueClef "treble" \stemUp d'8^"flute" c d e fis2
}
>>
d,,4 r a r
```



Notez que les deux instructions `\cueClef` et `\cueClefUnset` sont disponibles si nécessaire en dehors d'un `CueVoice`.

```
\clef "bass"
R1
\cueClef "treble"
d'8^"flute" c d e fis2
\cueClefUnset
d,,4 r a r
```



Lorsque la situation est plus complexe, instrument transpositeur ou citations de plusieurs sources, vous disposez des instructions `\cueDuring` et `\cueDuringWithClef`, versions spécifiques de la commande `\quoteDuring` – voir la rubrique précédente ([Citation d'autres voix], page 192).

Leur syntaxe est :

```
\cueDuring #origine #position #musique
```

et

```
\cueDuringWithClef #origine #position #clef #musique
```

Des mesures issues de la partie d'*origine* seront recopiées dans un contexte de `CueVoice` et synchronisées avec *musique* – habituellement un silence. L'apparition des petites notes initialise une polyphonie temporaire pour la portée concernée. L'argument *position* détermine si ces petites notes seront attachées à la première ou à la seconde voix – `UP` pour la première, `DOWN` pour la seconde.

```
fluteNotes = \relative c'' {
  r2. c4 | d8 c d e fis2 | g2 d |
}

oboeNotes = \relative c'' {
  R1
  \new CueVoice { \set instrumentCueName = "flute" }
  \cueDuring #"flute" #UP { R1 }
  g2 c,
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \oboeNotes
}
```



La propriété `quotedCueEventTypes` permet de définir précisément quels éléments de la voix originale seront reproduits. Sa valeur par défaut est `'(note-event rest-event tie-event beam-event tuplet-span-event)`. LilyPond reproduira donc les notes, silences, liaisons de prolongation, ligatures et nolets, mais pas les articulations, annotations ni nuances.

Note : Dans l'exemple précédent, il était nécessaire de déclarer explicitement le contexte `Voice`, sinon l'intégralité de l'expression musicale se serait retrouvée dans le contexte `CueVoice`.

```
oboeNotes = \relative c'' {
  r2 r8 d16(\f f e g f a)
  g8 g16 g g2.
}

\addQuote "oboe" { \oboeNotes }

\new Voice \relative c'' {
  \set Score.quotedCueEventTypes = #'(note-event rest-event tie-event
    beam-event tuplet-span-event
```

```

dynamic-event slur-event)

\cueDuring #"oboe" #UP { R1 }
g2 c,
}

```



Le nom de l'instrument qui est répliqué sera imprimé dès lors qu'aura été définie la propriété `instrumentCueName` d'un contexte `CueVoice` temporaire. Le positionnement et le style adopté par `instrumentCueName` dépendent de l'objet `\instrumentSwitch` – voir [Noms d'instrument], page 188. Par ailleurs, si la citation nécessite l'apparition d'une clef différente, celle-ci devra être introduite manuellement, tout comme l'originale qui devra être rappelée en fin de citation.

```

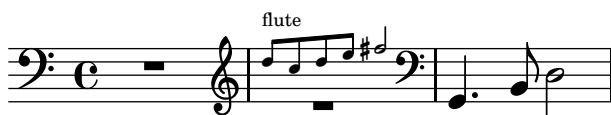
fluteNotes = \relative c'' {
  r2. c4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \clef treble
  \new CueVoice { \set instrumentCueName = "flute" }
  \cueDuring #"flute" #UP { R1 }
  \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}

```



L'instruction `\cueDuringWithClef` se chargera quant à elle, et grâce à un argument supplémentaire, de gérer le changement de clef nécessaire à la citation et le retour à la clef originelle.

```

fluteNotes = \relative c'' {
  r2. c4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \new CueVoice { \set instrumentCueName = "flute" }
  \cueDuringWithClef #"flute" #UP #"treble" { R1 }
}

```

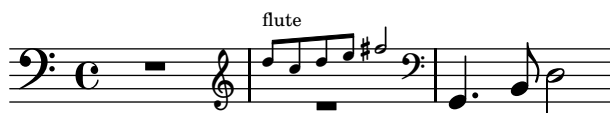
```

g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}

```



L'instruction `\cueDuring`, à l'instar de la commande `\quoteDuring`, tient compte des instruments transposeurs. La citation s'effectue aux hauteurs correspondant à l'instrument où elles apparaissent.

L'instruction `\transposedCueDuring` est particulièrement adaptée pour des instrument ayant une tessiture éloignée, comme dans le cas d'un piccolo cité dans une partie de clarinette basse. Sa syntaxe est identique à celle de `\cueDuring`, à ceci près qu'elle nécessite un argument supplémentaire afin de spécifier la transposition à effectuer en hauteur absolue.

```

piccoloNotes = \relative c''' {
  \clef "treble^8"
  R1
  c8 c c e g2
  c4 g g2
}

bassClarinetNotes = \relative c' {
  \key d \major
  \transposition bes,
  d4 r a r
  \transposedCueDuring #"piccolo" #UP d { R1 }
  d4 r a r
}

```

```

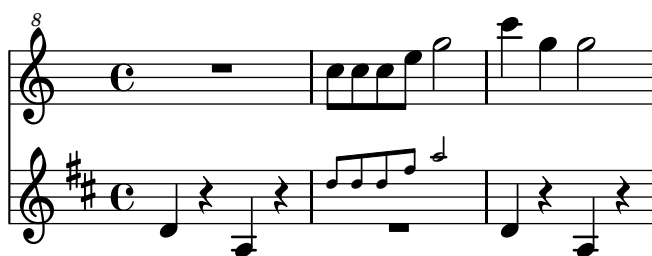
\addQuote "piccolo" { \piccoloNotes }

```

```

<<
  \new Staff \piccoloNotes
  \new Staff \bassClarinetNotes
>>

```



La commande `killCues` permet de supprimer les notes d'une citation. Ceci est utile lorsque cette citation n'est pas imprimée dans le conducteur entre autres. `killCues` supprimera les

notes et autres événements pris en charge par `\cueDuring`. Pour les autres annotations telles que changement de clef ou instrument concerné, faites appel à des balises – voir [Utilisation de balises], page 477 à ce sujet.

```
fluteNotes = \relative c' {
  r2. c4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \tag #'part {
    \clef treble
    \new CueVoice { \set instrumentCueName = "flute" }
  }
  \cueDuring #"flute" #UP { R1 }
  \tag #'part \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}

\new StaffGroup <<
  \new Staff {
    \fluteNotes
  }
  \new Staff {
    \removeWithTag #'part { \killCues { \bassoonNotes } }
  }
>>
```

The image shows a musical score for the song "The Rose Tree". It consists of two systems of music. The first system features a flute part in the upper staff and a vocal part in the lower staff. The flute part begins with a whole rest, followed by a melodic line: a quarter note G4, an eighth note A4, a quarter note B4, an eighth note C5, a quarter note D5, and a half note E5. The vocal part begins with a whole rest, followed by a half note G3, a quarter note A3, and a half note B3. The second system continues the vocal part with a half note C4, a quarter note D4, and a half note E4. The flute part has a whole rest in the first measure of the second system, followed by a half note G4, a quarter note A4, and a half note B4. The key signature has one sharp (F#), and the time signature is common time (C).

Lorsque la citation comporte une étiquette indiquant l'instrument qui joue et un changement de clef, ces éléments peuvent se regrouper à l'aide de la commande `\addInstrumentDefinition`. Pour plus d'information, reportez-vous au chapitre [\[Noms d'instrument\]](#), page 188.

Voir aussi

Manuel de notation : [Citation d’autres voix], page 192, [Instruments transpositeurs], page 22, [Noms d’instrument], page 188, [Clefs], page 16, [Citation-repère], page 281, [Utilisation de balises], page 477.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

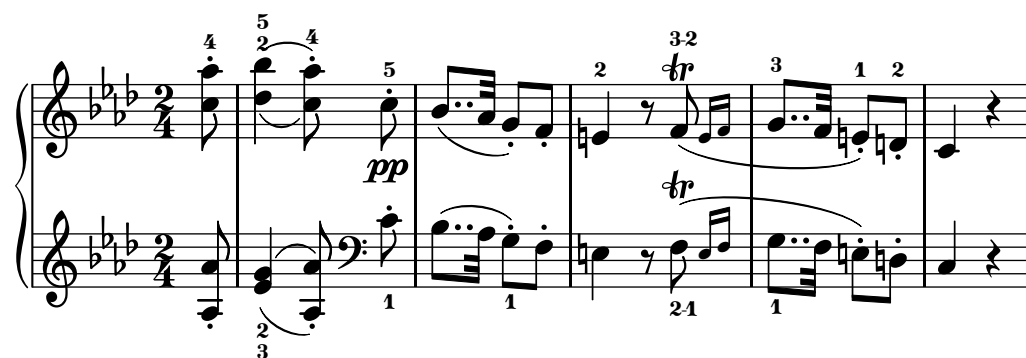
Référence des propriétés internes : Section “CueVoice” dans *Référence des propriétés internes*, Section “Voice” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

La commande `\cueDuring` ne sait pas gérer les collisions de silence entre les contextes `Voice` et `CueVoice`.

Dans le cadre d’un `\cueDuringWithClef` ou d’un `\transposedCueDuring`, l’argument supplémentaire doit intervenir après l’origine et la position.

1.7 Annotations éditoriales



Ce chapitre traite de la manière de modifier l’apparence des notes dans un but pédagogique ou d’analyse.

1.7.1 Dans la portée

Nous allons voir ici comment mettre en exergue certains éléments à l’intérieur même de la portée.

Indication de la taille de fonte musicale

Le plus sûr moyen de régler la taille de la police, quel que soit le contexte, consiste à définir la propriété `fontSize`. Ceci ne modifiera en rien la taille des différents symboles tels que ligatures ou liaisons.

Note : En matière de taille des éléments textuels, référez-vous au chapitre [Sélection de la fonte et de la taille], page 223.

```
\huge
c4.-> d8---3
\large
c4.-> d8---3
\normalsize
c4.-> d8---3
\small
c4.-> d8---3
\tiny
```

```
c4.-> d8---3
\teeny
c4.-> d8---3
```



En interne, la propriété `fontSize` d'un contexte aura pour effet de définir la propriété `font-size` pour tous les objets de rendu. La valeur de `font-size` est un entier représentant la taille proportionnellement à la hauteur standard de la portée en cours. Chaque incrément correspond à une augmentation d'environ 12 % de la taille de la police. Un pas de six aboutit exactement au doublement de la taille. La fonction Scheme `magstep` convertit le nombre affecté à `font-size` en facteur d'échelle. Vous pouvez aussi définir directement la propriété `font-size` de manière à n'affecter seulement que certains objets de rendu.

```
\set fontSize = #3
c4.-> d8---3
\override NoteHead #'font-size = #-4
c4.-> d8---3
\override Script #'font-size = #2
c4.-> d8---3
\override Stem #'font-size = #-5
c4.-> d8---3
```



Pour changer la taille des caractères, LilyPond met à l'échelle la fonte dont la taille est la plus proche de la taille voulue. La taille standard (pour laquelle `font-size = #0`) dépend de la hauteur de la portée. À une portée de 20 points correspond une police de 10 points.

La propriété `font-size` ne peut intervenir que pour les objets qui utilisent des polices, autrement dit ceux qui disposent de l'interface de rendu `font-interface`.

Commandes prédéfinies

`\teeny`, `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`.

Voir aussi

Morceaux choisis : [Section "Annotations éditoriales"](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section "font-interface"](#) dans *Référence des propriétés internes*.

Doigtés

Les doigtés peuvent être indiqués comme suit : `note-chiffre_du_doigt`

```
c4-1 d-2 f-4 e-3
```



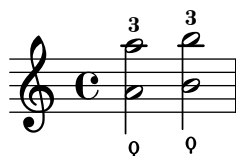
Pour les substitutions de doigts, on a recours à une indication textuelle (commande `\markup`) de doigté (commande `\finger`).

```
c4-1 d-2 f-4 c^\markup { \finger "2 - 3" }
```



La commande `\thumb` peut être utilisée pour indiquer, par exemple dans une partition de violoncelle, si une note doit être jouée avec le pouce (*thumb* en anglais).

```
<a_\thumb a'-3>2 <b_\thumb b'-3>
```



Les doigtés des accords peuvent être saisis note par note, en les indiquant après chaque hauteur de note.

```
<c-1 e-2 g-3 b-5>2 <d-1 f-2 a-3 c-5>
```



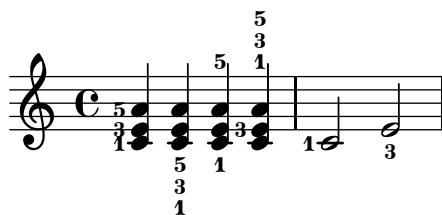
Les indications de doigtés peuvent se placer au-dessus ou en dessous de la portée – voir [Section 5.4.2 \[Direction et positionnement\]](#), page 579 à ce sujet.

Morceaux choisis

Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord `< >`, même s'il ne s'agit que d'une seule note.

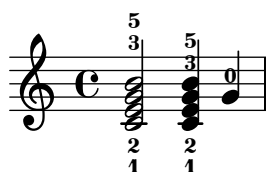
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
}
```



Impression des doigtés à l'intérieur de la portée

L'empilement des indications de doigté se fait par défaut à l'extérieur de la portée. Néanmoins, il est possible d'annuler ce comportement.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering #'staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 <g'-0>
}
```



Évitement de collision des doigtés d'un accord

Doigtés et numéros de corde, lorsque affectés à des notes individuelles, évitent les hampes et ligatures. Cette fonctionnalité n'est pas activée par défaut en ce qui concerne les notes appartenant à un accord. Voici comment mettre en place la dérogation appropriée en pareil cas.

```
\relative c' {
  \set fingeringOrientations = #'(up)
  \set stringNumberOrientations = #'(up)
  \set strokeFingerOrientations = #'(up)

  % Default behavior
  r8
  <f c'-5>8
  <f c'\5>8
  <f c'-\rightHandFinger #2 >8

  % Corrected to avoid collisions
  r8
  \override Fingering #'add-stem-support = ##t
  <f c'-5>8
  \override StringNumber #'add-stem-support = ##t
  <f c'\5>8
  \override StrokeFinger #'add-stem-support = ##t
  <f c'-\rightHandFinger #2 >8
}
```



Voir aussi

Manuel de notation : [Section 5.4.2 \[Direction et positionnement\]](#), page 579.

Morceaux choisis : [Section “Annotations éditoriales”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “FingeringEvent”](#) dans *Référence des propriétés internes*, [Section “fingering-event”](#) dans *Référence des propriétés internes*, [Section “Fingering-engraver”](#) dans *Référence des propriétés internes*, [Section “New-fingering-engraver”](#) dans *Référence des propriétés internes*, [Section “Fingering”](#) dans *Référence des propriétés internes*.

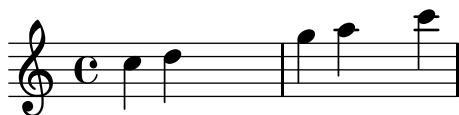
Problèmes connus et avertissements

La syntaxe ‘*note-digit*’ ne prend pas en charge, par défaut, les chiffres supérieurs à 9.

Dictée à trous

Les notes masquées – ou invisibles ou encore transparentes – sont utiles dans le cadre d’exercices de théorie ou de composition.

```
c4 d
\hideNotes
e4 f
\unHideNotes
g a
\hideNotes
b
\unHideNotes
c
```



Têtes de note, hampes, crochets et silences sont invisibles. Une ligature sera invisible si elle démarre sur une note invisible. Les objets de notation attachés à une note invisible ne seront pas masqués pour autant.

```
e8(\p f g a)--
\hideNotes
e8(\p f g a)--
```



Commandes prédéfinies

`\hideNotes`, `\unHideNotes`.

Voir aussi

Manuel d’initiation : [Section “Visibilité et couleur des objets”](#) dans *Manuel d’initiation*.

Manuel de notation : [\[Silences invisibles\]](#), page 53, [Section 5.4.6 \[Visibilité des objets\]](#), page 586, [\[Masquage de portées\]](#), page 185.

Morceaux choisis : [Section “Annotations éditoriales”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Note-spacing-engraver”](#) dans *Référence des propriétés internes*, [Section “NoteSpacing”](#) dans *Référence des propriétés internes*.

Coloration d'objets

Des objets peuvent être colorisés individuellement. Une listes des noms des couleurs disponibles se trouvent à l'annexe [Section A.7 \[Liste des couleurs\]](#), page 619.

```
\override NoteHead #'color = #red
c4 c
\override NoteHead #'color = #(x11-color 'LimeGreen)
d
\override Stem #'color = #blue
e
```



L'intégralité de la palette des couleurs définies pour X11 est accessible par la fonction Scheme `x11-color`. Cette fonction prend en argument une expression symbolique de la forme `'TaraTata` ou bien une chaîne de caractères comme `"TaraTata"`. La première formulation est à la fois plus rapide à écrire et aussi plus efficace. Néanmoins, la deuxième forme permet d'accéder aux noms composés des couleurs de X11.

Lorsque la fonction `x11-color` ne trouve pas le paramètre fourni, elle revient à la couleur par défaut, le noir. Le problème ressort de façon évidente au vu de la partition finale.

```
\override Staff.StaffSymbol #'color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}
```

```
gis8 a
\override Beam #'color = #(x11-color "medium turquoise")
gis a
\override Accidental #'color = #(x11-color 'DarkRed)
gis a
\override NoteHead #'color = #(x11-color "LimeGreen")
gis a
% this is deliberate nonsense; note that the stems remain black
\override Stem #'color = #(x11-color 'Boggle)
b2 cis
```



Un autre moyen consiste à fournir à la fonction Scheme `rgb-color` les composantes de la couleur exacte au format rouge-vert-bleu (*RGB*) – chacune étant exprimée en en fraction de 256 (le 0.5 de l'exemple suivant correspond à 128).

```
\override Staff.StaffSymbol #'color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}
```

```
\override Stem #'color = #(rgb-color 0 0 0)
gis8 a
```

```
\override Stem #'color = #(rgb-color 1 1 1)
gis8 a
\override Stem #'color = #(rgb-color 0 0 0.5)
gis4 a
```



Voir aussi

Manuel de notation : [Section A.7 \[Liste des couleurs\]](#), page 619, [Section 5.3.4 \[La commande d’affinage \(tweak\)\]](#), page 573.

Morceaux choisis : [Section “Annotations éditoriales”](#) dans *Morceaux choisis*.

Problèmes connus et avertissements

Une couleur x11 n’aura pas forcément le même rendu qu’une couleur normale ayant un nom similaire.

Les couleurs de X11 ne sont pas toutes perceptibles dans un navigateur internet. Aussi nous vous recommandons, dans le cadre d’une présentation multimédia, d’utiliser des couleurs de base `blue`, `green`, `red` – bleu, vert, rouge.

Vous ne pouvez pas coloriser des notes à l’intérieur d’un accord avec `\override`. Si besoin est, utilisez `\tweak`. Pour plus de détails, consultez [Section 5.3.4 \[La commande d’affinage \(tweak\)\]](#), page 573.

Parenthèses

Des objets peuvent être mis entre parenthèses en saisissant `\parenthesize` juste avant l’événement musical. Si l’instruction préfixe un accord, chaque note le composant se présentera entre parenthèses. Vous pouvez aussi mettre individuellement entre parenthèses les notes d’un accord.

```
c2 \parenthesize d
c2 \parenthesize <c e g>
c2 <c \parenthesize e g>
```



Les objets autres que des notes peuvent aussi être entre parenthèses. En ce qui concerne les articulations, l’instruction `\parenthesize` doit cependant être précédée d’un tiret.

```
c2-\parenthesize -. d
c2 \parenthesize r
```



Voir aussi

Morceaux choisis : [Section “Annotations éditoriales”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Parenthesis engraver”](#) dans *Référence des propriétés internes*, [Section “ParenthesesItem”](#) dans *Référence des propriétés internes*, [Section “parentheses-interface”](#) dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Lorsqu'un accord est mis entre parenthèses, celles-ci s'appliquent individuellement à chacune des notes le composant, alors qu'on attendrait une seule paire de parenthèses encadrant tout l'accord.

Hampes

Dès qu'une note est rencontrée, un objet `Stem` se crée automatiquement. Pour les rondes et les silences, ils sont aussi créés, mais en mode invisible.

L'orientation des hampes peut être définie manuellement – voir [Section 5.4.2 \[Direction et positionnement\]](#), page 579 à ce sujet.

Commandes prédéfinies

`\stemUp`, `\stemDown`, `\stemNeutral`.

Morceaux choisis

Direction par défaut des hampes de la ligne médiane

La direction des hampes des notes placées sur la ligne médiane de la portée est gérée par la propriété `neutral-direction` de l'objet `Stem`.

```
\relative c' ' {
  a4 b c b
  \override Stem #'neutral-direction = #up
  a4 b c b
  \override Stem #'neutral-direction = #down
  a4 b c b
}
```



Voir aussi

Manuel de notation : [Section 5.4.2 \[Direction et positionnement\]](#), page 579.

Morceaux choisis : [Section “Annotations éditoriales”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Stem_engraver”](#) dans *Référence des propriétés internes*, [Section “Stem”](#) dans *Référence des propriétés internes*, [Section “stem-interface”](#) dans *Référence des propriétés internes*.

1.7.2 Hors de la portée

Nous allons nous intéresser ici à souligner des éléments inscrits dans la portée par des éléments qui lui seront externes.

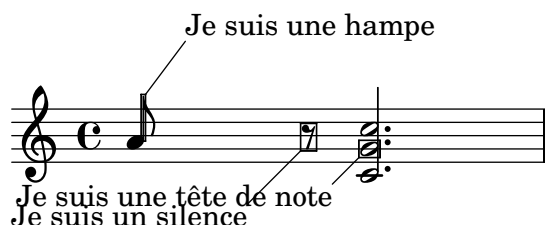
Info-bulle

Vous pouvez marquer et nommer des éléments de notation à l'aide de bulles. L'objectif premier de cette fonctionnalité est d'expliquer la notation.

En voici un exemple :

```
\new Voice \with { \consists "Balloon_engraver" }
{
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Je suis une hampe" }
  a8
}
```

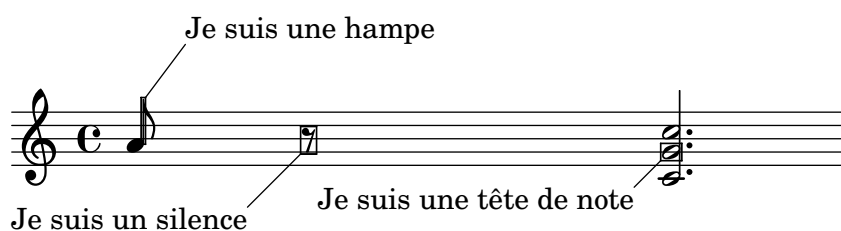
```
\balloonGrobText #'Rest #'(-4 . -4) \markup { "Je suis un silence" }
r
<c, g'-\balloonText #'(-2 . -2) \markup { "Je suis une tête de note" } c>2.
}
```



Vous disposez de deux fonctions musicales, `balloonGrobText` et `balloonText`. `balloonGrobText` prend en argument l'objet graphique à agrémenter et s'utilise comme `\once \override`. `balloonText`, quant à lui, s'utilise comme une simple articulation et fonctionne comme `\tweak` pour attacher du texte à l'une des notes d'un accord. Les autres arguments sont le décalage et le texte de la bulle.

Les info-bulles ont une influence sur les espacements, que l'on peut gérer ainsi :

```
\new Voice \with { \consists "Balloon_engraver" }
{
  \balloonLengthOff
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Je suis une hampe" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Je suis un silence" }
  r
  \balloonLengthOn
  <c, g'-\balloonText #'(-2 . -2) \markup { "Je suis une tête de note" } c>2.
}
```



Commandes prédéfinies

`\balloonLengthOn`, `\balloonLengthOff`.

Voir aussi

Morceaux choisis : [Section “Annotations éditoriales”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Balloon_engraver”](#) dans *Référence des propriétés internes*, [Section “BalloonTextItem”](#) dans *Référence des propriétés internes*, [Section “balloon-interface”](#) dans *Référence des propriétés internes*.

Quadrillage temporel

Vous pouvez tracer des lignes entre les portées, synchronisées avec les notes.

LilyPond a recours à deux graveurs distincts afin d'afficher le quadrillage : le `Grid_point_engraver` se charge de déterminer l'envergure du crochet, alors que le `Grid_line_span_engraver` se consacrera au tracé des lignes. Les lignes sont par défaut centrées horizontalement

sous les notes et alignées sur la gauche des têtes. La propriété `gridInterval` spécifie quant à elle l'espace de temps entre chaque ligne.

```
\layout {
  \context {
    \Staff
    \consists "Grid_point_engraver" %% active les guides
    gridInterval = #(ly:make-moment 1 4)
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    %% centre les lignes guides horizontalement sous les notes
  }
}

\score {
  \new ChoirStaff <<
    \new Staff \relative c'' {
      \stemUp
      c'4. d8 e8 f g4
    }
    \new Staff \relative c {
      %% centre les lignes guides verticalement
      \clef bass
      \stemDown
      c4 g' f e
    }
  }
  >>
}
```



Morceaux choisis

Apparence du quadrillage temporel

Modifier certaines des propriétés du quadrillage temporel aura pour effet d'en changer l'apparence.

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \relative c'' {
        \stemUp
        c'4. d8 e8 f g4
      }
    }
  }
}
```

```

}
\new Staff {
  \relative c {
    % this moves them up one staff space from the default position
    \override Score.GridLine #'extra-offset = #'(0.0 . 1.0)
    \stemDown
    \clef bass
    \once \override Score.GridLine #'thickness = #5.0
    c4
    \once \override Score.GridLine #'thickness = #1.0
    g'4
    \once \override Score.GridLine #'thickness = #3.0
    f4
    \once \override Score.GridLine #'thickness = #5.0
    e4
  }
}
>>
\layout {
  \context {
    \Staff
    % set up grids
    \consists "Grid_point_engraver"
    % set the grid interval to one quarter note
    gridInterval = #(ly:make-moment 1 4)
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % this moves them to the right half a staff space
    \override NoteColumn #'X-offset = #-0.5
  }
}
}

```



Voir aussi

Morceaux choisis : [Section “Annotations éditoriales”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Grid_line_span_engraver”](#) dans *Référence des propriétés internes*, [Section “Grid_point_engraver”](#) dans *Référence des propriétés internes*, [Section “GridLine”](#) dans *Référence des propriétés internes*, [Section “GridPoint”](#) dans *Référence des propriétés internes*, [Section “grid-line-interface”](#) dans *Référence des propriétés internes*, [Section “grid-point-interface”](#) dans *Référence des propriétés internes*.

Crochets d’analyse

On utilise des crochets en analyse musicale, pour indiquer la structure d’une pièce.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
  c2\startGroup
  d\stopGroup
}
```



Les crochets d’analyses sont susceptibles d’être imbriqués :

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
  c4\startGroup\startGroup
  d4\stopGroup
  e4\startGroup
  d4\stopGroup\stopGroup
}
```



Voir aussi

Morceaux choisis : [Section “Annotations éditoriales”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Horizontal_bracket_engraver”](#) dans *Référence des propriétés internes*, [Section “HorizontalBracket”](#) dans *Référence des propriétés internes*, [Section “horizontal-bracket-interface”](#) dans *Référence des propriétés internes*, [Section “Staff”](#) dans *Référence des propriétés internes*.

1.8 Texte

The image displays a musical score in 3/4 time, featuring piano and vocal staves. The piano part includes instructions such as *p con amabilità*, *ten.*, *tranqu. dolce*, *ten.*, and *ten.*. The vocal part includes *ten.* and *tr.*. Below the staves, there are several lines of text: *cantabile, con intimissimo sentimento, ma sempre molto dolce e semplice*, *non staccato*, *molto p, sempre tranquillo ed egualmente, non rubato*, and several instances of *Red.* (likely a placeholder for a redaction or a specific instruction).

Nous allons voir ici comment insérer du texte dans une partition, ainsi que différentes manières de le mettre en forme.

Certains éléments de texte ne sont pas abordés ici mais dans des chapitres qui leur sont dédiés. C'est le cas de la [Section 2.1 \[Musique vocale\]](#), page 238 et des [Section 3.2 \[Titres et entêtes\]](#), page 451.

1.8.1 Ajout de texte

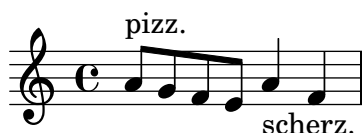
Cette partie constitue une introduction aux différentes manières d'ajouter du texte à une partition.

Note : Pour écrire des accents et autres caractères spéciaux, il suffit de les insérer directement dans votre fichier LilyPond. Ce fichier devra être sauvegardé avec l'encodage UTF-8. Pour plus d'informations, voir [\[Codage du texte\]](#), page 480.

Commentaires textuels

Vous pouvez ajouter à une partition des indications sous forme textuelle, comme dans l'exemple suivant. Ces indications se placeront manuellement au-dessus ou au-dessous de la portée selon la syntaxe utilisée – cf. [Section 5.4.2 \[Direction et positionnement\]](#), page 579.

```
a8^"pizz." g f e a4-"scherz." f
```



Cette syntaxe est en fait un raccourci. Des constructions plus élaborées d'annotation peuvent être obtenues en ayant recours à un bloc `\markup` et selon les préceptes énoncés dans [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

```
a8^\markup { \italic pizz. } g f e
a4_\markup { \tiny scherz. \bold molto } f
```



Par défaut, ces indications n'affectent en rien l'espacement des notes. Leur longueur peut néanmoins être prise en considération : dans l'exemple qui suit, le premier commentaire n'influe pas sur l'espacement, à l'inverse du second.

```
a8^"pizz." g f e
\textLength0n
a4_"scherzando" f
```



En plus d'indications textuelles, les notes peuvent se voir attacher des articulations, comme indiqué au chapitre [\[Articulations et ornements\]](#), page 109.

Pour de plus amples détails sur la manière de combiner indications textuelles et articulations, reportez-vous au chapitre [Section “Positionnement des objets”](#) dans *Manuel d'initiation*.

Commandes prédéfinies

```
\textLength0n, \textLength0ff.
```

Voir aussi

Manuel d'initiation : [Section “Positionnement des objets”](#) dans *Manuel d'initiation*.

Manuel de notation : [Section 1.8.2 \[Mise en forme du texte\]](#), page 221, [Section 5.4.2 \[Direction et positionnement\]](#), page 579, [\[Articulations et ornements\]](#), page 109.

Morceaux choisis : [Section “Texte”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “TextScript”](#) dans *Référence des propriétés internes*.

Problèmes connus et avertissements

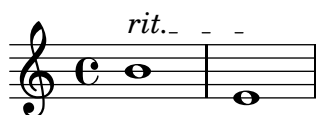
S'assurer que tous les éléments textuels et les paroles respectent les marges du document requiert des calculs relativement lourds ; vous pouvez toutefois vous en affranchir en ajoutant

```
\override Score.PaperColumn #'keep-inside-line = ##f
```

Indication textuelle avec extension

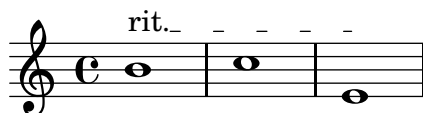
Certaines indications d'interprétation comme *rallentando*, *accelerando* ou *trilles*, s'inscrivent textuellement et se prolongent sur plusieurs notes à l'aide d'une ligne pleine, pointillée ou ondulée. Ces objets, que l'on appelle « extenseurs », se dessinent entre deux notes à l'aide de la syntaxe suivante :

```
\override TextSpanner #'(bound-details left text) = "rit."
b1\startTextSpan
e,\stopTextSpan
```



Le texte à imprimer est spécifié en tant que propriété de l'objet `TextSpanner`. Il apparaîtra par défaut en italique ; cependant, rien ne s'oppose à un autre graphisme dès lors que vous faites appel à un bloc `\markup` – voir [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

```
\override TextSpanner #'(bound-details left text) =
  \markup { \upright "rit." }
b1\startTextSpan c
e,\stopTextSpan
```



Le style de ligne se définit lui aussi comme une propriété de l'objet. Les détails concernant la syntaxe à utiliser sont expliqués au chapitre [Section 5.4.7 \[Styles de ligne\]](#), page 591.

Commandes prédéfinies

```
\textSpannerUp, \textSpannerDown, \textSpannerNeutral.
```

Morceaux choisis

Extensions de nuance postfix

Les lignes d'extension des commandes `\cresc`, `\dim` et `\decrec` peuvent désormais être personnalisées facilement sous forme d'opérateurs postfix. Soufflets et (de)crescendos peuvent cohabiter. `\<` et `\>` produiront par défaut des soufflets, alors que `\cresc` etc. produiront une indication textuelle avec extension.

```
% Some sample text dynamic spanners, to be used as postfix operators
crpoco =
#(make-music 'CrescendoEvent
  'span-direction START
  'span-type 'text
  'span-text "cresc. poco a poco")

\relative c' {
```



```

c4\cresc d4 e4 f4 |
g4 a4\! b4\crpoco c4 |
c4 d4 e4 f4 |
g4 a4\! b4< c4 |
g4\dim a4 b4\decreasc c4\!
}

```



Personnalisation des extenseurs de nuance postfix

Il s'agit de fonctions postfix pour personnaliser l'extension des crescendos textuels. L'extension devrait débiter sur la première note de la mesure. Il faut utiliser `-\mycresc` – comme une articulation – sous peine que le départ de l'extension n'apparaisse qu'à la note suivante.

```

% Two functions for (de)crescendo spanners where you can explicitly give the
% spanner text.

```

```

mycresc =

```

```

#(define-music-function (parser location mymarkup) (markup?)
  (make-music 'CrescendoEvent
    'span-direction START
    'span-type 'text
    'span-text mymarkup))

```

```

mydecresc =

```

```

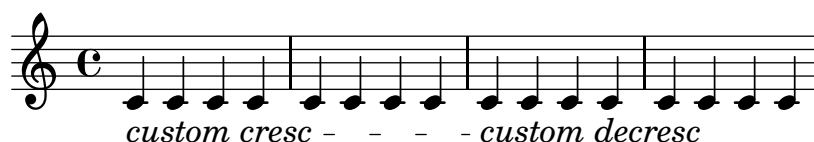
#(define-music-function (parser location mymarkup) (markup?)
  (make-music 'DecrescendoEvent
    'span-direction START
    'span-type 'text
    'span-text mymarkup))

```

```

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4\! c4 c4
}

```



Voir aussi

Manuel de notation : Section 5.4.7 [Styles de ligne], page 591, [Nuances], page 112, Section 1.8.2 [Mise en forme du texte], page 221.

Morceaux choisis : Section “Texte” dans *Morceaux choisis*, Section “Signes d’interprétation” dans *Morceaux choisis*.

Référence des propriétés internes : Section “TextSpanner” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

LilyPond ne peut traiter qu'un seul extenseur à la fois par voix.

Indications textuelles

La commande `\mark` est tout d'abord conçue pour les [\[Indications de repère\]](#), page 100.

```
c4
\mark "Allegro"
c c c
```



Cette syntaxe rend possible l'adjonction de n'importe quel texte à une barre de mesure. Ce texte peut être mis en forme de différentes manières dès lors qu'est utilisé un bloc `\markup`, comme indiqué au chapitre [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

```
<c e>1
\mark \markup { \italic { colla parte } }
<d f>2 <e g>
<c f aes>1
```



Elle peut aussi servir à insérer des signes de *coda* ou de *segno*, ou bien un point d'orgue, au-dessus d'une barre de mesure. Couplez-la alors à la commande `\markup` pour avoir accès au symbole approprié, selon les indications contenues au chapitre [\[Notation musicale dans du texte formaté\]](#), page 231.

```
<bes f>2 <aes d>
\mark \markup { \musicglyph #"scripts.ufermata" }
<e g>1
```



Le résultat de `\mark` n'apparaîtra que sur la portée supérieure d'un système. Si vous introduisez la commande `\mark` au moment d'une barre de mesure, la marque se placera au-dessus de la barre. Si vous y faites appel au milieu d'une mesure, cette marque sera positionnée entre les notes. Si elle intervient en début de ligne, elle sera placée juste avant la première note de cette portée. Enfin, une marque qui tomberait sur un saut de ligne sera imprimée au début de la ligne suivante.

```
\mark "Allegro"
c1 c
\mark "assai" \break
c c
```

Allegro**assai****Morceaux choisis**

Indication textuelle en fin de ligne

Les indications textuelles peuvent être imprimées à la fin d'une ligne plutôt qu'en tête de la suivante. L'alignement sur la barre de mesure devra alors s'effectuer par l'extrémité droite de l'indication.

```
\relative c' ' {
  g2 c
  d,2 a'
  \once \override Score.RehearsalMark #'break-visibility = #end-of-line-visible
  \once \override Score.RehearsalMark #'self-alignment-X = #RIGHT
  \mark "D.C. al Fine"
  \break
  g2 b,
  c1 \bar "||"
}
```



Alignement des indications par rapport à divers objets de notation

Les indications textuelles peuvent s'aligner par rapport à d'autres objets que des barres de mesure, tels que `ambitus`, `breathing-sign`, `clef`, `custos`, `staff-bar`, `left-edge`, `key-cancellation`, `key-signature` ou `time-signature`.

Par défaut, les indications textuelles sont alignées sur le milieu des objets de notation. Bien entendu, vous pouvez modifier les propriétés des objets en question pour obtenir un autre résultat comme l'illustre la deuxième ligne de l'exemple. Dans le cas de portées multiples, ces réglages doivent être faits pour chacune d'entre elles.

```
\relative c' {
  e1

  % the RehearsalMark will be centered above the Clef
  \override Score.RehearsalMark #'break-align-symbols = #'(clef)
  \key a \major
  \clef treble
  \mark \markup { \char ##x2193 }
  e1
```

```

% the RehearsalMark will be centered above the TimeSignature
\override Score.RehearsalMark #'break-align-symbols = #'(time-signature)
\key a \major
\clef treble
\time 3/4
\mark \markup { \char ##x2193 }
e2.

% the RehearsalMark will be centered above the KeySignature
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature)
\key a \major
\clef treble
\time 4/4
\mark \markup { \char ##x2193 }
e1

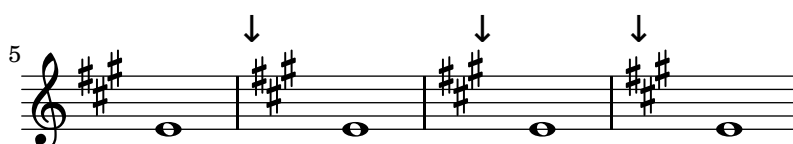
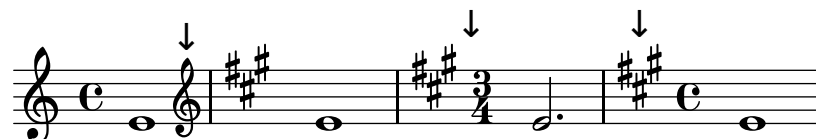
\break
e1

% the RehearsalMark will be aligned with the left edge of the KeySignature
\once \override Score.KeySignature #'break-align-anchor-alignment = #LEFT
\mark \markup { \char ##x2193 }
\key a \major
e1

% the RehearsalMark will be aligned with the right edge of the KeySignature
\once \override Score.KeySignature #'break-align-anchor-alignment = #RIGHT
\key a \major
\mark \markup { \char ##x2193 }
e1

% the RehearsalMark will be aligned with the left edge of the KeySignature
% and then shifted right by one unit.
\once \override Score.KeySignature #'break-align-anchor = #1
\key a \major
\mark \markup { \char ##x2193 }
e1
}

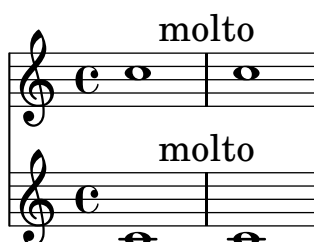
```



Impression des indications sur toutes les portées d'un système

Bien que ces indications textuelles ne soient habituellement imprimées qu'au niveau de la portée supérieure d'un système, leur affectation peut être répecutée à chacune des portées.

```
\score {
  <<
    \new Staff { c''1 \mark "molto" c'' }
    \new Staff { c'1 \mark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
    \context {
      \Staff
      \consists "Mark_engraver"
      \consists "Staff_collecting_engraver"
    }
  }
}
```

**Voir aussi**

Manuel de notation : [Indications de repère], page 100, Section 1.8.2 [Mise en forme du texte], page 221, [Notation musicale dans du texte formaté], page 231, Section A.8 [La fonte Feta], page 620.

Morceaux choisis: *Section “Texte” dans Morceaux choisis.*

Référence des propriétés internes : *Section “MarkEvent” dans Référence des propriétés internes, Section “Mark_engraver” dans Référence des propriétés internes, Section “RehearsalMark” dans Référence des propriétés internes.*

Texte indépendant

Un bloc `\markup` peut exister en lui-même, indépendamment de tout bloc `\score`, et venir en préambule par exemple – voir le chapitre *Section 3.1.5 [Structure de fichier], page 449* à ce propos.

```
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
```

Tomorrow, and tomorrow, and tomorrow...

De cette manière, vous pouvez insérer du texte en dehors de la musique. Ceci devient particulièrement utile lorsque le fichier source contient plusieurs morceaux. Pour plus d'informations à

ce propos, reportez-vous au chapitre [Section 3.1.2 \[Plusieurs partitions dans un même ouvrage\]](#), [page 446](#).

```
\score {
  c'1
}
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
\score {
  c'1
}
```



Tomorrow, and tomorrow, and tomorrow...



Les blocs de textes peuvent s'étendre sur plusieurs pages, ce qui permet de générer des ouvrages complets uniquement grâce à LilyPond. Cette fonctionnalité, ainsi que la syntaxe appropriée, est abordée plus en détail au chapitre [\[Texte avec sauts de page\]](#), [page 233](#).

Commandes prédéfinies

`\markup`, `\markuplist`.

Morceaux choisis

Bloc de texte indépendant sur deux colonnes

L'utilisation de la commande `\markup` permet de distribuer un bloc de texte indépendant sur plusieurs colonnes.

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { 0 sacrum convivium }
      \line { in quo Christus sumitur, }
      \line { recolitur memoria passionis ejus, }
      \line { mens impletur gratia, }
      \line { futurae gloriae nobis pignus datur. }
      \line { Amen. }
    }
  }
  \hspace #2
  \column {
    \line { \italic { 0 sacred feast } }
    \line { \italic { in which Christ is received, } }
    \line { \italic { the memory of His Passion is renewed, } }
    \line { \italic { the mind is filled with grace, } }
    \line { \italic { and a pledge of future glory is given to us. } }
  }
}
```

```

\line { \italic { Amen. } }
}
\hspace #1
}
}

```

O sacrum convivium	<i>O sacred feast</i>
in quo Christus sumitur,	<i>in which Christ is received,</i>
recolitur memoria passionis ejus,	<i>the memory of His Passion is renewed,</i>
mens impletur gratia,	<i>the mind is filled with grace,</i>
futurae gloriae nobis pignus datur.	<i>and a pledge of future glory is given to us.</i>
Amen.	<i>Amen.</i>

Voir aussi

Manuel de notation : [Section 1.8.2 \[Mise en forme du texte\]](#), page 221, [Section 3.1.5 \[Structure de fichier\]](#), page 449, [Section 3.1.2 \[Plusieurs partitions dans un même ouvrage\]](#), page 446, [\[Texte avec sauts de page\]](#), page 233.

Morceaux choisis : [Section “Texte” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “TextScript” dans *Référence des propriétés internes*](#).

1.8.2 Mise en forme du texte

Nous allons voir dans les lignes qui suivent la manière de mettre en forme du texte à l’aide de la syntaxe propre au mode `\markup`.

Introduction au formatage de texte

La commande `\markup` permet d’ajouter du texte et dispose de sa propre syntaxe que nous appellerons le « mode *markup* ».

La syntaxe du mode *markup* n’est pas différente de celle des autres modes de LilyPond : une expression `\markup` est bornée par des accolades `{ ... }`. Un mot unique sera considéré comme une expression minimale, et n’aura donc pas besoin d’être mis entre accolades.

Contrairement aux indications simples, du type "entre guillemets", les blocs `\markup` peuvent contenir des expressions imbriquées ou d’autres commandes *markup*, dès lors qu’elles sont précédées du caractère `\`. Ces commandes n’affecteront que la première expression qui les suit.

```

a1-\markup intenso
a2^\markup { poco \italic più forte }
c e1
d2_\markup { \italic "string. assai" }
e
b1^\markup { \bold { molto \italic agitato } }
c

```



Un bloc `\markup` peut contenir du texte entre guillemets. De telles chaînes seront considérées comme des expressions textuelles minimales ; à ce titre, toute commande de type *markup* ou

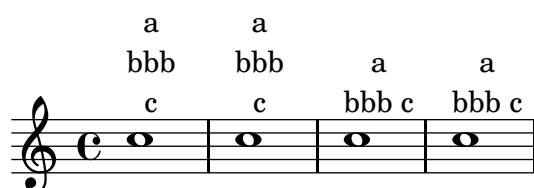
tout caractère spécial – tel un `\` ou un `#` – sera imprimé littéralement et sans influencer sur le formatage du texte. Il est de ce fait possible d'imprimer des guillemets informatiques " dès lors qu'ils sont précédés d'une oblique inverse.

```
a1^\italic markup..."
a_\markup { \italic "... imprime des lettres en \"italique\" !" }
a a
```



Une liste de mots, pour pouvoir être traitée en tant qu'expression distincte, doit être bornée par des " ou précédée d'une commande. La manière de définir les expressions *markup* aura une influence sur la manière dont elles seront empilées, centrées ou alignées. Dans l'exemple qui suit, la deuxième expression `\markup` est traitée tout comme la première :

```
c1^\markup { \center-column { a bbb c } }
c1^\markup { \center-column { a { bbb c } } }
c1^\markup { \center-column { a \line { bbb c } } }
c1^\markup { \center-column { a "bbb c" } }
```



Vous pouvez stocker les étiquettes textuelles en tant que variables, et attacher ces identificateurs à des notes, comme ici :

```
allegro = \markup { \bold \large Allegro }

{
  d''8.^allegro
  d'16 d'4 r2
}
```



Pour une liste des différentes commandes spécifiques au mode `\markup`, consultez l'annexe [\(undefined\)](#) [Text markup commands], page [\(undefined\)](#) (en anglais).

Voir aussi

Manuel de notation : [\(undefined\)](#) [Text markup commands], page [\(undefined\)](#).

Fichiers d'initialisation : `'scm/markup.scm'`.

Morceaux choisis : [Section "Texte" dans Morceaux choisis.](#)

Problèmes connus et avertissements

Les erreurs de syntaxe en mode *markup* sont peu explicites.

Sélection de la fonte et de la taille

Le mode *markup* autorise des changements élémentaires de la fonte :

```
d1^\markup {
  \bold { Più mosso }
  \italic { non troppo \underline Vivo }
}
r2 r4 r8
d,_\markup { \italic quasi \smallCaps Tromba }
f1 d2 r
```



La taille des caractères se modifie, relativement à la taille globale des portées, de différentes manières.

Vous pouvez adopter l'une des tailles prédéfinies, comme ici :

```
b1_\markup { \huge Sinfonia }
b1^\markup { \teeny da }
b1-\markup { \normalsize camera }
```



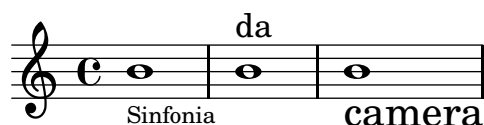
Vous pouvez la modifier relativement à sa valeur précédente :

```
b1_\markup { \larger Sinfonia }
b1^\markup { \smaller da }
b1-\markup { \magnify #0.6 camera }
```



Vous pouvez l'augmenter ou la diminuer par rapport à la taille globale de portée :

```
b1_\markup { \fontsize #-2 Sinfonia }
b1^\markup { \fontsize #1 da }
b1-\markup { \fontsize #3 camera }
```



Vous pouvez lui attribuer une valeur arbitraire quelle que soit la taille de portée globale :

```
b1_\markup { \abs-fontsize #20 Sinfonia }
b1^\markup { \abs-fontsize #8 da }
b1-\markup { \abs-fontsize #14 camera }
```



Vous pouvez imprimer du texte en indice ou en exposant. Celui-ci sera dans une taille plus petite, mais rien ne s'oppose à ce que vous lui affectiez une taille normale :

```
\markup {
  \column {
    \line { 1 \super st movement }
    \line { 1 \normal-size-super st movement }
    \sub { (part two) } }
}
```

1st movement
1st movement (part two)

Le mode *markup* vous permet de sélectionner d'autres familles de fontes. Par défaut, LilyPond utilise une police avec empattement, du type roman, et tout changement doit être explicite. Dans la dernière ligne de l'exemple qui suit, vous noterez qu'il n'y a aucune différence entre les premier et deuxième mots.

```
\markup {
  \column {
    \line { Act \number 1 }
    \line { \sans { Scene I. } }
    \line { \typewriter { Verona. An open place. } }
    \line { Enter \roman Valentine and Proteus. }
  }
}
```

Act 1

Scene I.

Verona. An open place.

Enter Valentine and Proteus.

Certaines familles de police spécifiques aux nombres ou aux nuances par exemple, ne disposent pas de tous les caractères, comme nous l'avons vu dans les chapitres [\[Personnalisation des indications de nuance\]](#), page 117 et [\[Indications de reprise manuelles\]](#), page 140.

Lorsqu'un changement survient au milieu d'un mot, il se peut qu'un espacement supplémentaire apparaisse. Il suffit en pareil cas de concaténer les différents éléments :

```
\markup {
  \column {
    \line {
      \concat { 1 \super st }
      movement
    }
    \line {
      \concat { \dynamic p , }
      \italic { con dolce espressione }
    }
  }
}
```

1st mouvement

***p**, con dolce espressione*

Une liste des différentes commandes permettant de changer de fonte ou d'utiliser des fontes personnalisées est disponible à l'annexe [Section A.10.1 \[Font\]](#), page 642.

Pour savoir comment personnaliser des fontes, reportez-vous au chapitre [Section 1.8.3 \[Fontes\]](#), page 234.

Commandes prédéfinies

`\teeny`, `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`, `\smaller`, `\larger`.

Voir aussi

Manuel de notation : [Section A.10.1 \[Font\]](#), page 642, [\[Personnalisation des indications de nuance\]](#), page 117, [\[Indications de reprise manuelles\]](#), page 140, [Section 1.8.3 \[Fontes\]](#), page 234.

Fichiers d'initialisation : `'scm/define-markup-commands.scm'`.

Morceaux choisis : [Section “Texte” dans Morceaux choisis](#).

Référence des propriétés internes : [Section “TextScript” dans Référence des propriétés internes](#).

Problèmes connus et avertissements

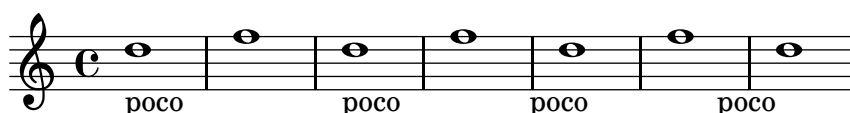
Le recours aux commandes `\teeny`, `\tiny`, `\small`, `\normalsize`, `\large` et `\huge` produiront des espacements nettement moins réguliers que si vous utilisez `\fontsize`.

Alignement du texte

Cette partie traite de la manière de positionner du texte en mode *markup*. On déplace l'intégralité d'un objet *markup* en utilisant la syntaxe décrite au chapitre [Section “Déplacement d'objets” dans Manuel d'initiation](#).

Les objets de type *markup* peuvent s'aligner de différentes manières. Une indication textuelle est par défaut alignée sur son extrémité gauche. Dans l'exemple qui suit, il n'y a aucune différence entre les deux premiers *markups*.

```
d1-\markup { poco }
f
d-\markup { \left-align poco }
f
d-\markup { \center-align { poco } }
f
d-\markup { \right-align poco }
```



L'alignement horizontal peut être ajusté à l'aide d'une valeur numérique :

```
a1-\markup { \halign #-1 poco }
e'
a,-\markup { \halign #0 poco }
e'
a,-\markup { \halign #0.5 poco }
e'
a,-\markup { \halign #2 poco }
```



Certains objets possèdent leurs propres procédures d'alignement, qui annuleront toute spécification d'alignement que vous pourriez leur affecter. La solution consiste alors à déplacer l'intégralité de ces objets *markup*, comme indiqué par exemple au chapitre [Indications textuelles], page 216.

L'alignement vertical est quant à lui un peu plus compliqué. Comme nous l'avons vu ci-avant, les objets *markup* peuvent être déplacés dans leur intégralité. Il est néanmoins possible de déplacer certains éléments spécifiques au sein d'un bloc *markup*. En pareil cas, l'élément à déplacer doit être précédé d'un « point d'ancrage » – un autre élément du *markup* ou un objet invisible. L'exemple qui suit illustre ces deux possibilités. Vous noterez par ailleurs que le dernier *markup*, dépourvu de point d'ancrage, n'est de ce fait pas déplacé.

```
d2^\markup {
  Acte I
  \raise #2 { Scène 1 }
}
a'
g_\markup {
  \null
  \lower #4 \bold { Très modéré }
}
a
d,^\markup {
  \raise #4 \italic { Une forêt. }
}
a'4 a g2 a
```



Très modéré

Certaines commandes permettent de régler l'alignement des objets textuels en mode *markup*, tant au niveau horizontal que vertical. Tout objet soumis à ces commandes doit être précédé d'un point d'ancrage.

```
d2^\markup {
  Acte I
  \translate #'(-1 . 2) "Scène 1"
}
a'
g_\markup {
  \null
  \general-align #Y #3.2 \bold "Très modéré"
}
a
d,^\markup {
  \null
  \translate-scaled #'(-1 . 2) \teeny "Une forêt."
}
a'4 a g2 a
```

**Très modéré**

Un objet de type *markup* peut contenir plusieurs lignes de texte. Dans l'exemple suivant, chaque élément ou expression ira se placer sur sa propre ligne, tantôt alignée à gauche, tantôt centrée.

```
\markup {
  \column {
    a
    "b c"
    \line { d e f }
  }
  \hspace #10
  \center-column {
    a
    "b c"
    \line { d e f }
  }
}
```

a	a
b c	b c
d e f	d e f

Pareillement, une liste d'éléments ou d'expressions sera répartie sur une ligne entière, voire même centrée sur toute la page s'il n'y a qu'un seul élément. De telles expressions peuvent à leur tour contenir du texte multiligne ou une autre expression *markup*.

```
\markup {
  \fill-line {
    \line { William S. Gilbert }
    \center-column {
      \huge \smallCaps "The Mikado"
      or
      \smallCaps "The Town of Titipu"
    }
    \line { Sir Arthur Sullivan }
  }
}
```

```
\markup {
  \fill-line { 1885 }
}
```

William S. Gilbert

THE MIKADO
or
THE TOWN OF TITIPU

Sir Arthur Sullivan

1885

Les indications textuelles, lorsqu'elles sont relativement longues, peuvent se répartir sur plusieurs lignes en fonction de la largeur de ligne. Le texte sera alors soit aligné à gauche, soit justifié, comme le montre l'exemple suivant :

```

\markup {
  \column {
    \line \smallCaps { La vida breve }
    \line \bold { Acto I }
    \wordwrap \italic {
      (La escena representa el corral de una casa de
      gitanos en el Albaicín de Granada. Al fondo una
      puerta por la que se ve el negro interior de
      una Fragua, iluminado por los rojos resplandores
      del fuego.)
    }
    \hspace #0

    \line \bold { Acto II }
    \override #'(line-width . 50)
    \justify \italic {
      (Calle de Granada. Fachada de la casa de Carmela
      y su hermano Manuel con grandes ventanas abiertas
      a través de las que se ve el patio
      donde se celebra una alegre fiesta)
    }
  }
}

```

LA VIDA BREVE

Acto I

(La escena representa el corral de una casa de gitanos en el Albaicín de Granada. Al fondo una puerta por la que se ve el negro interior de una Fragua, iluminado por los rojos resplandores del fuego.)

Acto II

(Calle de Granada. Fachada de la casa de Carmela y su hermano Manuel con grandes ventanas abiertas a través de las que se ve el patio donde se celebra una alegre fiesta)

Une liste des différentes commandes permettant d'aligner du texte en mode *markup* est disponible à l'annexe [Section A.10.2 \[Align\]](#), page 650.

Voir aussi

Manuel d'initiation : [Section “Déplacement d'objets”](#) dans *Manuel d'initiation*.

Manuel de notation : [Section A.10.2 \[Align\]](#), page 650, [\[Indications textuelles\]](#), page 216.

Fichiers d'initialisation : `'scm/define-markup-commands.scm'`.

Morceaux choisis : [Section “Texte”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “TextScript”](#) dans *Référence des propriétés internes*.

Éléments graphiques dans du texte formaté

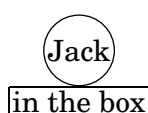
Vous pouvez, grâce aux mode *markup*, ajouter divers objets graphiques à votre partition.

Certaines commandes de *markup* permettent d'ornementer des éléments textuels avec des graphismes, à l'instar de l'exemple suivant :

```

\markup \fill-line {
  \center-column {
    \circle Jack
    \box "in the box"
    \null
    \line {
      Erik Satie
      \hspace #3
      \bracket "1866 - 1925"
    }
    \null
    \rounded-box \bold Prelude
  }
}

```



Erik Satie [1866 - 1925]

Prelude

Certaines directives peuvent nécessiter d'accroître l'espacement autour du texte – voir l'annexe [Section A.10.2 \[Align\], page 650](#) pour une liste des différentes commandes particulières au mode *markup* ainsi que leur description.

```

\markup \fill-line {
  \center-column {
    \box "Charles Ives (1874 - 1954)"
    \null
    \box \pad-markup #2 "THE UNANSWERED QUESTION"
    \box \pad-x #8 "A Cosmic Landscape"
    \null
  }
}
\markup \column {
  \line {
    \hspace #10
    \box \pad-to-box #'(-5 . 20) #'(0 . 5)
    \bold "Largo to Presto"
  }
  \pad-around #3
  "String quartet keeps very even time,
  Flute quartet keeps very uneven time."
}

```

Charles Ives (1874 - 1954)

THE UNANSWERED QUESTION

A Cosmic Landscape

Largo to Presto

String quartet keeps very even time, Flute quartet keeps very uneven time.

Vous pouvez imprimer certains graphismes ou symboles sans qu'il n'y ait de texte. Ces objets peuvent même se combiner, à l'instar de n'importe quelle expression *markup*.

```
\markup {
  \combine
    \draw-circle #4 #0.4 ##f
    \filled-box #'(-4 . 4) #'(-0.5 . 0.5) #1
  \hspace #5

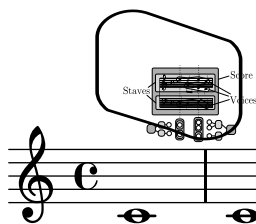
  \center-column {
    \triangle ##t
    \combine
      \draw-line #'(0 . 4)
      \arrow-head #Y #DOWN ##f
  }
}
```



Des fonctionnalités graphiques avancées vous permettent même d'inclure dans une partition un fichier image converti au format PostScript encapsulé (extension **eps**), ou bien de tracer un graphique directement dans le fichier source à partir d'instructions PostScript pures. Nous vous conseillons, en pareil cas, de toujours spécifier les dimensions du dessin, comme dans ce qui suit :

```
c1^\markup {
  \combine
    \epsfile #X #10 #"./context-example.eps"
    \with-dimensions #'(0 . 6) #'(0 . 10)
    \postscript #"
      -2 3 translate
      2.7 2 scale
      newpath
      2 -1 moveto
      4 -2 4 1 1 arct
      4 2 3 3 1 arct
      0 4 0 3 1 arct
      0 0 1 -1 1 arct
      closepath
      stroke"
}
```


c



L'annexe [Section A.10.3 \[Graphic\]](#), page 665 répertorie les différentes commandes en matière de graphisme.

Voir aussi

Manuel de notation : [Section A.10.3 \[Graphic\]](#), page 665, [Section 1.7 \[Annotations éditoriales\]](#), page 200, [Section A.10.2 \[Align\]](#), page 650.

Fichiers d'initialisation : 'scm/define-markup-commands.scm', 'scm/stencil.scm'.

Morceaux choisis : [Section "Texte"](#) dans *Morceaux choisis*.

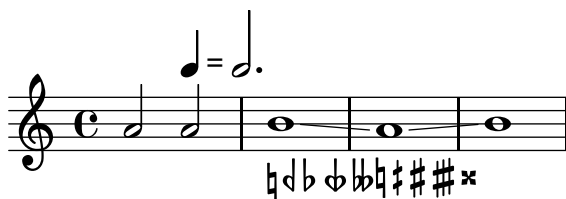
Référence des propriétés internes : [Section "TextScript"](#) dans *Référence des propriétés internes*.

Notation musicale dans du texte formaté

Divers éléments de notation peuvent orner une partition, au moyen d'un objet *markup*.

Notes et altérations sont données à l'aide d'instructions *markup* :

```
a2 a^\markup {
  \note #"4" #1
  =
  \note-by-number #1 #1 #1.5
}
b1_\markup {
  \natural \semiflat \flat
  \sesquiflat \doubleflat
}
\glissando
a1_\markup {
  \natural \semisharp \sharp
  \sesquisharp \doublesharp
}
\glissando b
```



Le mode *markup* permet d'accéder à d'autres objets de notation :

```
g1 bes
ees-\markup {
  \finger 4
  \tied-lyric #"~"
```

```

\finger 1
}
fis_\markup { \dynamic rf }
bes^\markup {
  \beam #8 #0.1 #0.5
}
cis
d-\markup {
  \markalphabet #8
  \markletter #8
}

```

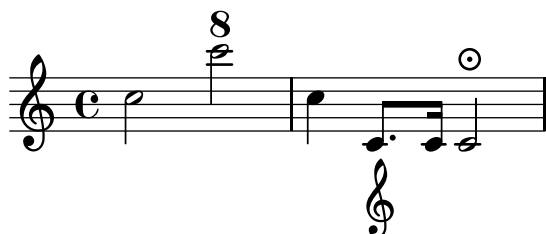


En règle générale, tout symbole musical peut être inclus dans un *markup*, comme le montre l'exemple qui suit. Ces différents symboles sont répertoriés dans l'annexe [Section A.8 \[La fonte Feta\]](#), page 620.

```

c2
c'^\markup { \musicglyph #"eight" }
c,4
c,8._\markup { \musicglyph #"clefs.G_change" }
c16
c2^\markup { \musicglyph #"timesig.neomensural94" }

```



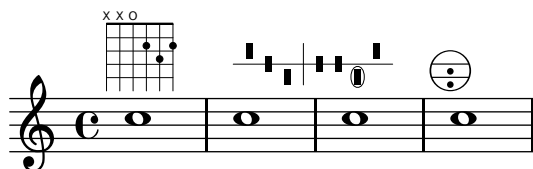
La rubrique [\[Tout savoir sur les fontes\]](#), page 234 contient d'autres informations sur l'impression de glyphes non alphabétiques, tels que des crochets ou accolades.

Le mode *markup* supporte aussi les diagrammes spécifiques à certains instruments :

```

c1^\markup {
  \fret-diagram-terse #"x;x;o;2;3;2;"
}
c^\markup {
  \harp-pedal #"^~v|--ov^"
}
c
c^\markup {
  \combine
    \musicglyph #"accordion.discant"
  \combine
    \raise #0.5 \musicglyph #"accordion.dot"
    \raise #1.5 \musicglyph #"accordion.dot"
}

```



La documentation sur ces diagrammes se trouvent à l'annexe [Section A.10.5 \[Instrument Specific Markup\]](#), page 675.

Rien ne s'oppose à ce qu'une étiquette ne comporte une partition. Il suffit que l'expression en question contienne un bloc `\score` incluant un bloc `\layout`.

```
c4 d^\markup {
  \score {
    \relative c' { c4 d e f }
    \layout { }
  }
}
e f |
c d e f
```



Les différentes commandes *markup* relatives à la notation musicale sont répertoriées à l'annexe [Section A.10.4 \[Music\]](#), page 671.

Voir aussi

Manuel de notation : [Section A.10.4 \[Music\]](#), page 671, [Section A.8 \[La fonte Feta\]](#), page 620, [\[Tout savoir sur les fontes\]](#), page 234.

Fichiers d'initialisation : `'scm/define-markup-commands.scm'`, `'scm/fret-diagrams.scm'`, `'scm/harp-pedals.scm'`.

Morceaux choisis : [Section "Texte"](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section "TextScript"](#) dans *Référence des propriétés internes*.

Texte avec sauts de page

Alors que `\markup` s'utilise pour traiter un bloc de texte insécable, `\markuplist` permet, employé en tête de partition, d'obtenir un bloc de lignes réparties différemment et, le cas échéant, sur plusieurs pages.

```
\markuplist {
  \justified-lines {
    Un long texte constitué de lignes justifiées.
    ...
  }
  \wordwrap-lines {
    Un autre grand paragraphe.
    ...
  }
  ...
}
```

Un long texte constitué de lignes justifiées. ...

Un autre grand paragraphe. ...

...

Cette syntaxe prend en charge une liste de *markups* ; il peut s'agir

- d'une suite de commandes générant à leur tour des lignes de texte,
- d'une liste de lignes de texte,
- d'une liste d'étiquettes.

Les différentes commandes permettant de générer des listes de lignes se trouve dans l'annexe [\(undefined\)](#) [Text markup list commands], page [\(undefined\)](#).

Commandes prédéfinies

`\markuplist.`

Voir aussi

Manuel de notation : [\(undefined\)](#) [Text markup list commands], page [\(undefined\)](#),

Manuel d'extension : [Section “Définition d’une nouvelle commande de liste de markups”](#) dans *Extension de LilyPond*.

Fichiers d'initialisation : `'scm/define-markup-commands.scm'`.

Morceaux choisis : [Section “Texte”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “TextScript”](#) dans *Référence des propriétés internes*.

1.8.3 Fontes

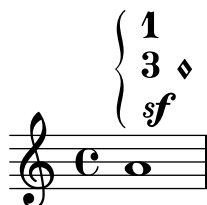
Ce chapitre est consacré aux fontes et polices de caractère, à leur gestion. Vous y apprendrez aussi comment changer de fonte en cours de partition.

Tout savoir sur les fontes

La gestion des fontes est assurée par plusieurs bibliothèques : FontConfig se charge de répertorier les différentes fontes installées sur votre système ; quant à Pango, elle se charge plus particulièrement de leur rendu.

Les fontes musicales peuvent se décrire comme un jeu de glyphes spécifiques regroupés dans plusieurs familles. L'exemple qui suit montre la syntaxe à utiliser pour accéder, en mode *markup*, aux différentes fontes *feta* non textuelles de LilyPond.

```
a1^\markup {
  \vcenter {
    \override #'(font-encoding . fetaBraces)
    \lookup #"brace120"
    \override #'(font-encoding . fetaText)
    \column { 1 3 sf }
    \override #'(font-encoding . fetaMusic)
    \lookup #"noteheads.s0petrucci"
  }
}
```



Tous ces glyphes, à l'exception des accolades qui sont regroupées dans **fetaBraces**, sont accessibles avec la syntaxe indiquée dans [Notation musicale dans du texte formaté], page 231.

Une remarque s'impose au sujet des glyphes contenus dans **fetaBraces** : chacun d'eux est formé du mot *brace* auquel est accolé un numéro d'ordre. Lorsque vous désirez imprimer une accolade, vous devez la « chercher » par son numéro d'ordre – d'où la fonction `\lookup` de l'exemple ci-dessus –, tout en sachant qu'il est compris entre 0 (la plus petite) et 575 (la plus grande). Vous serez souvent amené à procéder par tâtonnement pour arriver au résultat optimal. Par ailleurs, **fetaBraces** ne comporte que des accolades ouvrantes. Pour obtenir une accolade fermante, il suffit d'appliquer une rotation au glyphe sélectionné, comme indiqué à la rubrique Section 5.4.8 [Rotation des objets], page 592.

Vous disposez de trois familles de fontes textuelles : *roman* pour la police sérif ou avec empattement – fixée par défaut à New Century Schoolbook –, une police sans empattement (*sans* sérif) et une à chasse fixe (ou monospace) – les deux dernières étant déterminées par l'installation de Pango.

Chaque famille dispose en principe de différents styles et niveaux de gras. L'exemple qui suit illustre la manière de changer la famille, le style, la graisse ou la taille. Notez bien que l'argument fourni à `font-size` correspond à la correction à apporter à la taille par défaut.

```
\override Score.RehearsalMark #'font-family = #'typewriter
\mark \markup "Ouverture"
\override Voice.TextScript #'font-shape = #'italic
\override Voice.TextScript #'font-series = #'bold
d2.^ \markup "Allegro"
\override Voice.TextScript #'font-size = #-3
c4^smaller
```



Cette syntaxe s'applique aussi en mode *markup*, bien que celui-ci dispose d'une syntaxe allégée comme nous l'avons vu dans [Sélection de la fonte et de la taille], page 223 :

```
\markup {
  \column {
    \line {
      \override #'(font-shape . italic)
      \override #'(font-size . 4)
      Idomeneo,
    }
    \line {
      \override #'(font-family . typewriter)
      {
        \override #'(font-series . bold)
        re
        di
      }
      \override #'(font-family . sans)
      Creta
    }
  }
}
```

Idomeneo, re di Creta

En plus de pouvoir jongler entre les différentes fontes prédéfinies, LilyPond vous permet d'en utiliser d'autres, ce qui fait l'objet des deux prochaines parties : [\[Attribution d'une fonte en particulier\]](#), page 236 et [\[Choix des fontes par défaut\]](#), page 236.

Voir aussi

Manuel de notation : [Section A.8 \[La fonte Feta\]](#), page 620, [\[Notation musicale dans du texte formaté\]](#), page 231, [Section 5.4.8 \[Rotation des objets\]](#), page 592, [\[Sélection de la fonte et de la taille\]](#), page 223, [Section A.10.1 \[Font\]](#), page 642.

Attribution d'une fonte en particulier

Vous pouvez utiliser n'importe quelle police installée sur votre système, dès lors qu'elle est accessible par Fontconfig et que vous respectez la syntaxe suivante :

```
\override Staff.TimeSignature #'font-name = #"Bitstream Charter"
\override Staff.TimeSignature #'font-size = #2
\time 3/4
```

```
a1_\markup {
  \override #'(font-name . "Vera Bold")
    { Vera Bold }
}
```



Pour obtenir la liste de toutes les polices disponibles sur votre machine, lancez
`lilypond -dshow-available-fonts toto`
 (quel qu'il soit, le dernier argument est obligatoire).

Voir aussi

Manuel de notation : [\[Tout savoir sur les fontes\]](#), page 234, [\[Choix des fontes par défaut\]](#), page 236.

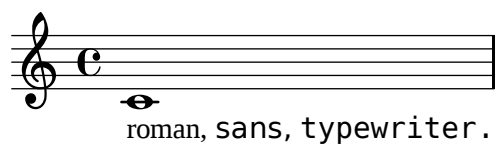
Morceaux choisis : [Section "Texte" dans *Morceaux choisis*](#).

Choix des fontes par défaut

Vous pouvez tout à fait préférer un autre jeu de polices par défaut que celui de LilyPond. Il vous faudra alors spécifier les différentes familles, en respectant l'ordre *roman*, *sans empattement* et *monospace*, comme dans l'exemple suivant. Pour plus d'explications sur les fontes, relisez [\[Tout savoir sur les fontes\]](#), page 234.

```
\paper {
  myStaffSize = #20
  #(define fonts
    (make-pango-font-tree "Times New Roman"
                          "Nimbus Sans"
                          "Luxi Mono"
                          (/ myStaffSize 20)))
}
```

```
\relative c'{
  c1-\markup {
    roman,
    \sans sans,
    \typewriter typewriter. }
}
```



Voir aussi

Manuel de notation : [Tout savoir sur les fontes], page 234, [Attribution d'une fonte en particulier], page 236, [Sélection de la fonte et de la taille], page 223, Section A.10.1 [Font], page 642.

2 Notation spécialisée

Ce chapitre explique comment créer la notation musicale spécifique à certains instruments ou certaines époques.

2.1 Musique vocale

Recitativo
Baritono

216

O Freun - de, nicht die - se Töne!

222

Sondern laßt uns an - ge -

228

neh-me-re an - stim-men, und freu -

232

denvollere!

ad libitum

Ce chapitre traite de la musique vocale : comment la saisir et comment s’assurer que les paroles s’alignent avec les notes de la mélodie correspondante.

2.1.1 Vue d’ensemble de la musique vocale

En complément de généralités, ce sous-chapitre aborde quelques styles particuliers en terme de musique vocale.

Références en matière de musique vocale

Graver de la musique vocale soulève plusieurs problèmes ; ils sont abordés soit dans ce chapitre, soit dans d’autres parties de la documentation de LilyPond.

- La plupart du temps, les paroles ne sont constituées que de texte simple. Cette forme de notation est abordée dans [Section “Écriture de chants simples” dans *Manuel d’initiation*](#).
- La musique vocale nécessite souvent de recourir au mode `markup`, aussi bien pour des paroles que pour d’autres éléments textuels comme le nom des personnages. Cette syntaxe est expliquée dans [\[Introduction au formatage de texte\]](#), page 221.
- L’impression d’un *ambitus* – ou tessiture – que l’on trouve en tête de certaines partitions, est abordée dans [\[Ambitus\]](#), page 30.
- Les indications de nuance viennent, par défaut, se placer sous la portée. Il en va différemment pour la musique vocale, de telle sorte qu’elles ne soient pas mélangées avec les paroles. Ceci fait l’objet de la rubrique [\[Mise en forme d’une partition chorale\]](#), page 276.

Voir aussi

Glossaire musicologique : [Section “ambitus” dans *Glossaire*](#).

Manuel d’initiation : [Section “Écriture de chants simples” dans *Manuel d’initiation*](#).

Manuel de notation : [\[Introduction au formatage de texte\]](#), page 221, [\[Ambitus\]](#), page 30, [\[Mise en forme d’une partition chorale\]](#), page 276.

Morceaux choisis : [Section “Musique vocale” dans *Morceaux choisis*](#).

Saisie des paroles

Il existe un mode de saisie spécialement adapté aux paroles. On l’introduit avec le mot-clé `\lyricmode`, ou en utilisant `\addlyrics` ou `\lyricsto`. Ce mode vous permet de saisir des paroles ainsi que leur ponctuation, de telle sorte que le caractère `a` ne sera plus interprété comme une note, un *la* pour les non latinistes, mais comme une syllabe. Les syllabes sont saisies comme des notes, mais les hauteurs sont alors remplacées par du texte. Exemple avec une comptine anglaise :

```
\lyricmode { Three4 blind mice,2 three4 blind mice2 }
```

Il y a deux manières principales de préciser la place exacte des syllabes : soit en spécifiant explicitement la durée de chaque syllabe – comme dans l’exemple ci-dessus – soit en alignant automatiquement les paroles sur les notes d’une mélodie ou d’une voix en utilisant `\addlyrics` ou `\lyricsto`. La première méthode est abordée plus en détail à la rubrique [\[Durée explicite des syllabes\]](#), page 245, et la deuxième à la rubrique [\[Durée automatique des syllabes\]](#), page 242.

Dans les paroles, un mot ou une syllabe commence par une lettre de l’alphabet, et se termine par une espace. Toute syllabe doit donc être séparée d’une autre par une espace, tout autre caractère – chiffre ou ponctuation – étant considéré comme partie intégrante de cette même syllabe. L’exemple suivant comporte une faute de frappe évidente :

```
\lyricmode { lah lah lah}
```

la dernière syllabe contient une `}` ; il y a de fait un défaut de parité avec l’accolade ouvrante, et la compilation échouera fort probablement. Prenez dès à présent l’habitude de toujours encadrer d’espaces une accolade :

```
\lyricmode { lah lah lah }
```

De la même manière, un point concluant une suite de lettres sera partie intégrante de la chaîne résultante. Par conséquent, il est impératif d’insérer des espaces lorsque vous modifiez les propriétés d’une commande. En d’autres termes, **ne saisissez pas**

```
\override Score.LyricText #'font-shape = #'italic
```

mais plutôt

```
\override Score . LyricText #'font-shape = #'italic
```

Pour utiliser des lettres accentuées ou des caractères spéciaux – cœurs ou guillemets inversés par exemple – il suffit de les insérer dans le fichier et de veiller à le sauvegarder avec le codage UTF-8. Voir à ce sujet [Section 3.3.3 \[Caractères spéciaux\]](#), page 480 pour plus de détails.

```
\relative c' { d8 c16 a bes8 f e' d c4 }
```

```
\addlyrics { „Schad’ um das schöne -- ne grü -- ne Band, }
```



Pour utiliser des guillemets informatiques standard, faites-les précéder d’une barre oblique inverse et encadrez d’une paire de guillemets la syllabe ainsi composée :

```
\relative c' { \time 3/4 e4 e4. e8 d4 e d c2. }
\addlyrics { "\"I" am so lone- "ly\""} said she }
```



Expliquer exactement comment LilyPond repère le début d'un mot en mode paroles (*Lyrics*) est quelque peu compliqué. En mode *Lyrics*, un mot peut commencer par : tout caractère alphabétique, `_`, `?`, `!`, `:`, `'`, un des codes de contrôle `^A` à `^F` et `^Q` à `^W`, `^Y`, `^^`, tout caractère ASCII de code strictement supérieur à 127, ou enfin l'un des symboles ```, `'`, `"`, ou `^`, s'il est précédé d'une barre oblique inverse.

LilyPond permet de contrôler très finement le rendu des paroles grâce au mode `\markup`, utilisable y compris au sein du mode `\lyricmode`. Des explications complètes sont disponibles au chapitre [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

Morceaux choisis

Mise en forme individuelle de syllabes

Le mode *markup* permet d'individualiser la mise en forme de certaines syllabes.

```
mel = \relative c'' { c4 c c c }
lyr = \lyricmode {
  Lyrics \markup { \italic can } \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}

<<
  \new Voice = melody \mel
  \new Lyrics \lyricsto melody \lyr
>>
```



Voir aussi

Manuel d'initiation : [Section “Chansons” dans Manuel d'initiation](#)

Manuel de notation : [Section 3.3.3 \[Caractères spéciaux\]](#), page 480, [\[Durée automatique des syllabes\]](#), page 242, [\[Durée explicite des syllabes\]](#), page 245, [Section 1.8.3 \[Fontes\]](#), page 234, [Section 1.8.2 \[Mise en forme du texte\]](#), page 221, [Section 5.4.1 \[Modes de saisie\]](#), page 577.

Référence des propriétés internes : [Section “LyricText” dans Référence des propriétés internes](#).

Alignement des paroles sur la mélodie

L'impression des paroles est réalisée à partir de l'interprétation d'un contexte *Lyrics* – voir [Section 5.1.1 \[Tout savoir sur les contextes\]](#), page 549.

```
\new Lyrics \lyricmode { ... }
```

Vous disposez de deux méthodes pour aligner des paroles sur une mélodie :

- Les paroles peuvent s'aligner automatiquement, la durée des syllabes étant déterminée à partir d'un contexte de voix ou, dans certaines circonstances, une mélodie associée, grâce aux commandes `\addlyrics` et `\lyricsto` ou en définissant la propriété `associatedVoice`. Ceci est détaillé à la rubrique [\[Durée automatique des syllabes\]](#), page 242.

```
<<
\new Staff <<
  \time 2/4
  \new Voice = "one" \relative c'' {
    \voiceOne
    c4 b8. a16 g4. r8 a4 ( b ) c2
  }
  \new Voice = "two" \relative c' {
    \voiceTwo
    s2 s4. f8 e4 d c2
  }
  }
>>

% takes durations and alignment from notes in "one"
\new Lyrics \lyricsto "one" {
  Life is __ _ love, live __ life.
}

% takes durations and alignment from notes in "one" initially
% then switches to "two"
\new Lyrics \lyricsto "one" {
  No more let
  \set associatedVoice = "two" % must be set one syllable early
  sins and sor -- rows grow.
}
>>
```



La première ligne de paroles est saisie selon la méthode la plus simple.

Vous pouvez constater, dans la deuxième ligne, que les paroles s'alignent selon les durées d'une voix différente. Ceci est particulièrement utile lorsque le texte s'agence différemment selon les couplets et que les durées sont accessibles grâce à des contextes `Voice` particuliers. Pour de plus amples détails, rendez-vous à la rubrique [Section 2.1.3 \[Couplets\]](#), page 267.

- Les paroles s'aligneront indépendamment de la valeur des notes dès lors que vous utiliserez le mode `\lyricmode` et affecterez explicitement leur durée à chaque syllabe.

```
<<
\new Voice = "one" \relative c'' {
  \time 2/4
  c4 b8. a16 g4. f8 e4 d c2
}
>>
```

```
% uses previous explicit duration of 2;
\new Lyrics \lyricmode {
  Joy to the earth!
}

% explicit durations, set to a different rhythm
\new Lyrics \lyricmode {
  Life4 is love,2. live4 life.2
}
>>
```



La première ligne de paroles ne s’aligne pas vraiment sur les notes parce qu’aucune durée n’a été spécifiée. En fait, LilyPond adopte la dernière durée mentionnée, un 2, et l’applique à chaque mot.

La deuxième ligne illustre la manière d’aligner des paroles sans tenir compte de la durée des notes. Cette façon de procéder permet de traiter un alignement différent selon les couplets lorsqu’il n’y a pas moyen de déduire les durées à partir d’un contexte musical ; la rubrique [\[Durée explicite des syllabes\]](#), page 245 aborde ceci plus en détails. Cette technique permet aussi d’ajouter des dialogues, comme indiqué à la rubrique [\[Dialogue et musique\]](#), page 285.

Des paroles saisies de cette manière s’aligneront par défaut sur la gauche des notes ; elles seront centrées sur les notes de la mélodie dès lors que vous pourrez les associer à une voix. Tout ceci est abordé plus en détails à la rubrique [\[Durée explicite des syllabes\]](#), page 245.

Voir aussi

Manuel d’initiation : [Section “Alignement des paroles sur une mélodie”](#) dans *Manuel d’initiation*.

Référence des propriétés internes : [Section “Lyrics”](#) dans *Référence des propriétés internes*.

Durée automatique des syllabes

Les paroles peuvent être automatiquement alignées sous une mélodie, de trois manières différentes :

- en utilisant la commande `\lyricsto` pour spécifier le contexte de voix qui contient la mélodie,
- en introduisant les paroles par la commande `\addlyrics`, placée juste après le contexte `Voice` qui contient la mélodie,
- en définissant la propriété `associatedVoice` pour synchroniser les paroles avec un autre contexte de voix, ce à n’importe quel moment.

Ces trois méthodes permettent d’ajouter les traits d’union séparant les syllabes d’un même mot ainsi que d’indiquer la tenue de la dernière syllabe. Ceci fait l’objet de la rubrique [\[Traits d’union et de prolongation\]](#), page 250.

Le contexte `Voice` contenant la mélodie sur laquelle les paroles vont s’aligner doit rester actif, au risque de voir la suite du texte disparaître. Ceci peut se produire lorsqu’il y a des moments où l’on ne chante pas. La rubrique [Section 5.1.3 \[Conservation d’un contexte\]](#), page 552 vous indiquera comment maintenir un contexte actif.

Utilisation de `\lyricsto`

Vous pouvez aligner automatiquement des paroles sous une mélodie en spécifiant à l'aide de la commande `\lyricsto` le contexte de voix qui contient cette mélodie :

```
<<
  \new Voice = "melodie" {
    a4 a a a
  }
  \new Lyrics \lyricsto "melodie" {
    Ce sont les mots
  }
>>
```



Cette commande adapte les paroles aux notes de la voix (contexte `Voice` dans le jargon Lily-Pond) *melodie*. Ce contexte `Voice` doit exister avant l'affectation des paroles par `\lyricsto`. La commande `\lyricsto` introduit automatiquement le mode `\lyricmode` ; il n'est alors pas nécessaire de rajouter `\lyricmode`. Les paroles viendront par défaut se placer en dessous des notes ; la rubrique [\[Positionnement vertical des paroles\]](#), page 252 vous donnera des indications pour d'autres options.

Utilisation de `\addlyrics`

La commande `\addlyrics` n'est en fait qu'une manière plus aisée d'écrire de la musique vocale dans une structure Lilypond plus complexe.

```
{ MUSIQUE }
\addlyrics { PAROLES }
revient au même que
\new Voice = "blah" { MUSIQUE }
\new Lyrics \lyricsto "blah" { PAROLES }
  En voici un exemple :
```

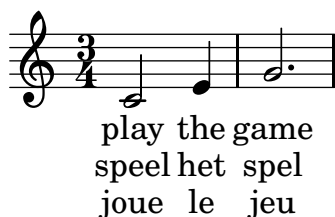
```
{
  \time 3/4
  \relative c' { c2 e4 g2. }
  \addlyrics { play the game }
}
```



On peut ajouter davantage de couplets en multipliant le nombre de sections `\addlyrics`.

```
{
  \time 3/4
  \relative c' { c2 e4 g2. }
  \addlyrics { play the game }
  \addlyrics { speel het spel }
  \addlyrics { joue le jeu }
}
```

}



Cependant, la commande `\addlyrics` ne peut pas gérer les constructions polyphoniques. Dans ce cas, mieux vaut employer `\lyricsto`.

Utilisation de `associatedVoice`

La propriété `associatedVoice` permet de basculer de mélodie pour la synchronisation des paroles. Elle s'emploie de la manière suivante :

```
\set associatedVoice = #"lala"
```

La valeur que vous attribuez à cette propriété (ici "lala") doit désigner un contexte `Voice` nommé, sans quoi les mélismes ne seront pas imprimés correctement.

Voici un exemple de cette manière de procéder :

```
<<
\new Staff <<
  \time 2/4
  \new Voice = "one" \relative c'' {
    \voiceOne
    c4 b8. a16 g4. r8 a4 ( b ) c2
  }
  \new Voice = "two" \relative c' {
    \voiceTwo
    s2 s4. f8 e8 d4. c2
  }
  }
>>
% takes durations and alignment from notes in "one" initially
% then switches to "two"
\new Lyrics \lyricsto "one">{
  No more let
  \set associatedVoice = "two" % must be set one syllable early
  sins and sor -- rows grow.
}
>>
```



Voir aussi

Manuel de notation : [Traits d'union et de prolongation], page 250, Section 5.1.3 [Conservation d'un contexte], page 552.

Durée explicite des syllabes

On peut aussi se passer de `\addlyrics`, `\lyricsto` et `associatedVoice` pour saisir des paroles. Dans ce cas, les syllabes sont entrées comme des notes – du texte remplaçant les hauteurs – ce qui veut dire que vous devez définir leur durée explicitement.

Par défaut, les syllabes seront alignées par la gauche sur l’instant musical. Les traits d’union seront imprimés entre les syllabes, à l’inverse des mélismes puisqu’il n’y a pas de voix associée.

Voici deux illustrations de cette technique :

```
<<
\new Voice = "melody" {
  \time 3/4
  c2 e4 g2 f
}
\new Lyrics \lyricmode {
  play1 the4 game4
}
>>
```



```
<<
\new Staff {
  \relative c'' {
    c2 c2
    d1
  }
}
\new Lyrics {
  \lyricmode {
    I2 like4. my8 cat!1
  }
}
\new Staff {
  \relative c' {
    c8 c c c c c c c
    c8 c c c c c c c
  }
}
>>
```



Cette manière de procéder est tout à fait adaptée lorsqu’un fond musical accompagne des dialogues – voir [\[Dialogue et musique\]](#), page 285.

Les syllabes seront centrées par rapport à l’instant musical dès lors que vous aurez attribué à la propriété `associatedVoice` le nom du contexte `Voice` contenant les notes. Grâce à l’utilisation de `associatedVoice`, les doubles tirets `--` ou soulignés `_` seront rendus correctement en trait d’union ou indication de mélisme.

```
<<
\new Voice = "melody" {
  \time 3/4
  c2 e4 g f g
}
\new Lyrics \lyricmode {
  \set associatedVoice = #"melody"
  play2 the4 game2. __
}
>>
```



Voir aussi

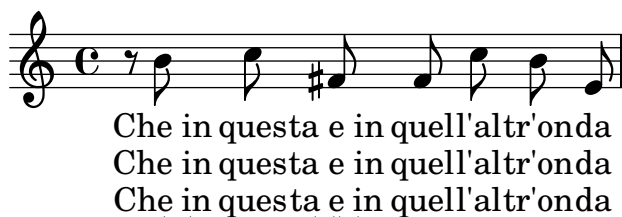
Manuel de notation : [\[Dialogue et musique\]](#), page 285.

Référence des propriétés internes : [Section “Lyrics”](#) dans *Référence des propriétés internes*, [Section “Voice”](#) dans *Référence des propriétés internes*.

Plusieurs syllabes sur une note

Pour attribuer plus d’une syllabe à une même note, vous pouvez soit les mettre entre guillemets, soit utiliser le caractère souligné (`_`) pour obtenir une espace, ou bien encore utiliser un tilde (`~`) pour obtenir une liaison entre les syllabes.

```
{
  \relative c' { \autoBeamOff
    r8 b c fis, fis c' b e, }
  \addlyrics { Che_in ques --ta_e_in quel -- l'al -- tr'on -- da }
  \addlyrics { "Che in" ques -- "ta e in" quel -- l'al -- tr'on -- da }
  \addlyrics { Che~in ques -- ta~e~in quel -- l'al -- tr'on -- da }
}
```



Voir aussi

Référence des propriétés internes : [Section “LyricCombineMusic”](#) dans *Référence des propriétés internes*.

Plusieurs notes pour une même syllabe

Parfois, tout particulièrement dans la musique médiévale, plusieurs notes correspondent à une même syllabe. Ces vocalises sont appelées **Section “mélismes” dans *Glossaire***. La syllabe à vocaliser est traditionnellement alignée sur la gauche de la première note du mélisme.

Lorsqu’un mélisme tombe sur une syllabe autre que la dernière d’un mot, un trait d’union étiré, indiqué par un double tiret -- dans le fichier source, reliera cette syllabe à la suivante.

Lorsqu’un mélisme tombe sur la dernière syllabe d’un mot ou que ce mot n’en comporte qu’une, l’usage est d’indiquer la « tenue » jusqu’à la dernière note de la vocalise. Ceci s’obtient en ajoutant un double caractère souligné __ après cette syllabe.

Vous disposez de cinq méthodes pour indiquer la présence d’un mélisme :

- Une liaison de prolongation constitue de fait un mélisme :

```
<<
\new Voice = "melody" {
  \time 3/4
  f4 g2 ~ |
  g4 e2 ~ |
  e8
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e __
}
>>
```



- LilyPond considère une liaison d’articulation comme un mélisme – il s’étendra de la première à la dernière note couvertes par cette liaison. Il s’agit là de la façon traditionnelle de saisir des paroles :

```
<<
\new Voice = "melody" {
  \time 3/4
  f4 g8 ( f e f )
  e8 ( d e2 )
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e __
}
>>
```



Notez bien qu’une liaison de phrasé – indiquée par `\(...\)` – n’a aucune incidence sur la gestion des mélismes.

- LilyPond considère des notes regroupées par une ligature manuelle comme un mélisme, si tant est que la procédure de ligature automatique a été désactivée – voir [\[Définition des règles de ligature automatique\]](#), page 78.

```
<<
  \new Voice = "melody" {
    \time 3/4
    \autoBeamOff
    f4 g8[ f e f]
    e2.
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e
  }
>>
```



Ceci ne peut, vous en conviendrez, prendre en compte des durées plus longues que la croche.

- LilyPond considère un groupe de notes non liées, mais encadrées par `\melisma` et `\melismaEnd`, comme constituant un mélisme :

```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g8
    \melisma
    f e f
    \melismaEnd
    e2.
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e
  }
>>
```



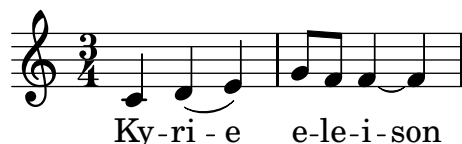
- Vous pouvez indiquer un mélisme directement dans les paroles, à l'aide d'un caractère souligné simple `_`, pour chaque note faisant partie de la vocalise :

```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g8 f e f
    e8 d e2
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- _ _ _ e _ _ _
  }
>>
```



Vous pouvez totalement désactiver l'interprétation des liaisons de prolongation ou d'articulation et des ligatures apparaissant dans une mélodie comme fait générateur d'un mélisme. Il suffit en ce cas de définir `melismaBusyProperties` :

```
<<
\new Voice = "melody" {
  \time 3/4
  \set melismaBusyProperties = #'()
  c4 d ( e )
  g8 [ f ] f4 ~ f
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e e -- le -- i -- son
}
>>
```



Certains réglages de `melismaBusyProperties` permettent de prendre en compte ou non les liaisons de tenue, les liaisons d'articulation et les ligatures dans la détection automatique des mélismes – voir `melismaBusyProperties` à la rubrique [Section “Tunable context properties”](#) dans *Référence des propriétés internes*.

Dans le cas où les indications de mélisme doivent être totalement ignorées, il vous faudra alors activer `ignoreMelismata` – voir [\[Rythme différent selon le couplet\]](#), page 269.

Lorsque, dans un passage où la propriété `melismaBusyProperties` est active, survient un mélisme, vous pouvez l'indiquer dans les paroles par un simple caractère souligné pour chaque note à inclure :

```
<<
\new Voice = "melody" {
  \time 3/4
  \set melismaBusyProperties = #'()
  c4 d ( e )
  g8 [ f ] ~ f4 ~ f
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- _ e _ _ _ _
}
>>
```



Commandes prédéfinies

`\autoBeamOff`, `\autoBeamOn`, `\melisma`, `\melismaEnd`.

Voir aussi

Glossaire musicologique : [Section “melisma” dans *Glossaire*](#).

Manuel d’initiation : [Section “Alignement des paroles sur une mélodie” dans *Manuel d’initiation*](#).

Manuel de notation : [\[Alignement des paroles sur la mélodie\]](#), page 240, [\[Durée automatique des syllabes\]](#), page 242, [\[Définition des règles de ligature automatique\]](#), page 78, [\[Rythme différent selon le couplet\]](#), page 269.

Référence des propriétés internes : [Section “Tunable context properties” dans *Référence des propriétés internes*](#).

Problèmes connus et avertissements

Certains mélismes ne sont pas détectés automatiquement ; vous devrez alors prolonger vous-même les syllabes concernées à l’aide d’un double caractère souligné.

Traits d’union et de prolongation

Un mélisme sur la dernière syllabe d’un mot est indiqué par une longue ligne horizontale basse s’étirant jusqu’à syllabe suivante. Une telle ligne, que nous appellerons prolongateur ou extenseur, s’obtient en saisissant ‘ `--` ’ (notez les espaces entourant le double caractère souligné).

Note : Dans une partition, les mélismes, ou vocalises, sont matérialisés par une ligne de prolongation. On l’indique par un double caractère souligné. Lorsqu’ils sont assez courts, ces mélismes peuvent s’indiquer par un souligné unique, ce qui aura pour effet de sauter une note à chaque fois et de ne pas imprimer de ligne.

Un trait d’union séparant deux syllabes d’un même mot s’obtient en saisissant ‘ `--` ’ (notez les espaces entourant le tiret double). Ce trait d’union sera centré entre les deux syllabes et sa longueur sera proportionnelle à l’espace les séparant.

Dans les partitions très serrées, les traits d’union peuvent ne pas être imprimés. Cet inconvénient peut être contrôlé par les propriétés `minimum-distance` pour gérer l’espace minimum entre deux syllabes, et `minimum-length`, seuil en deçà duquel il n’y a pas de trait d’union, toutes deux attachées à l’objet `LyricHyphen`.

Voir aussi

Référence des propriétés internes : [Section “LyricExtender” dans *Référence des propriétés internes*](#), [Section “LyricHyphen” dans *Référence des propriétés internes*](#)

2.1.2 Situations particulières en matière de paroles

Travail avec des paroles et variables

Vous pouvez créer des variables pour contenir les paroles, dès lors que vous faites appel au mode approprié :

```
musicOne = \relative c'' {
  c4 b8. a16 g4. f8 e4 d c2
}
verseOne = \lyricmode {
  Joy to the world, the Lord is come.
}
\score {
  <<
  \new Voice = "one" {
```

```

\time 2/4
\musicOne
}
\new Lyrics \lyricsto "one" {
  \verseOne
}
>>
}

```



La fonction `\lyricmode` permet de définir une variable pour les paroles. Point n'est besoin de spécifier les durées si vous utilisez `\addlyrics` ou `\lyricsto` lorsque vous y faites référence.

Pour une organisation différente ou plus complexe, mieux vaut commencer par créer et alimenter les variables contenant mélodies et paroles, puis définir la hiérarchie des portées et des lignes de paroles, et enfin combiner correctement mélodies et paroles à l'aide de la commande `\context`. Vous serez ainsi assuré que la voix à laquelle il est fait référence par `\lyricsto` aura bien été préalablement définie, comme dans l'exemple suivant :

```

sopranoMusic = \relative c'' { c4 c c c }
contraltoMusic = \relative c'' { a4 a a a }
sopranoWords = \lyricmode { Sop -- ra -- no words }
contraltoWords = \lyricmode { Con -- tral -- to words }

```

```

\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \sopranoMusic
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos"
    \new Staff {
      \new Voice = "contraltos" {
        \contraltoMusic
      }
    }
    \context Lyrics = "sopranos" {
      \lyricsto "sopranos" {
        \sopranoWords
      }
    }
    \context Lyrics = "contraltos" {
      \lyricsto "contraltos" {
        \contraltoWords
      }
    }
  }
  >>
}

```



Voir aussi

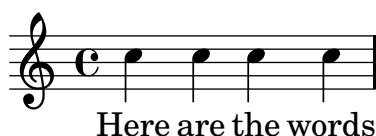
Manuel de notation : [Positionnement vertical des paroles], page 252.

Référence des propriétés internes : Section “LyricCombineMusic” dans *Référence des propriétés internes*, Section “Lyrics” dans *Référence des propriétés internes*.

Positionnement vertical des paroles

Selon le type de musique, les paroles apparaîtront au-dessus ou au-dessous d’une portée ou bien entre deux portées. Positionner des paroles en dessous de la portée à laquelle elles se rattachent est de loin la chose la plus simple : il suffit de mentionner le contexte de paroles après le contexte de portée :

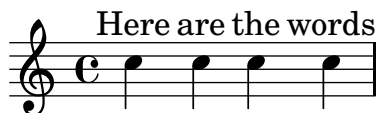
```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' { c4 c c c }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```



Positionner les paroles au-dessus de la portée se fait de deux manières différentes, le plus simple étant d’utiliser la même syntaxe que ci-dessus, à ceci près que la ligne de paroles sera positionnée de manière explicite :

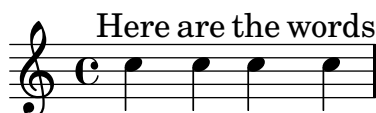
```
\score {
  <<
    \new Staff = "staff" {
      \new Voice = "melody" {
        \relative c'' { c4 c c c }
      }
    }
    \new Lyrics \with { alignAboveContext = "staff" } {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```

```
>>
}
```



Autre façon de procéder, cette fois-ci en deux étapes. Nous commençons par déclarer un contexte **Lyrics** que nous laissons vide, puis les contextes **Staff** et **Voice**. Dans un deuxième temps, nous ajoutons l'instruction `\context` et la commande `\lyricsto` pour affecter les paroles au contexte de voix en question. Voici comment cela se présente :

```
\score {
  <<
    \new Lyrics = "lyrics" \with {
      % lyrics above a staff should have this override
      \override VerticalAxisGroup #'staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "melody" {
        \relative c'' { c4 c c c }
      }
    }
    \context Lyrics = "lyrics" {
      \lyricsto "melody" {
        Here are the words
      }
    }
  }
  >>
}
```



Lorsque deux voix sont isolées chacune sur une portée, vous pouvez placer les paroles entre les deux portées en utilisant l'une des méthodes que nous venons de voir. En voici un exemple, basé sur la deuxième méthode :

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \relative c'' { c4 c c c }
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos" \with {
      % lyrics above a staff should have this override
      \override VerticalAxisGroup #'staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "contraltos" {
```

```

        \relative c'' { a4 a a a }
    }
}
\context Lyrics = "sopranos" {
    \lyricsto "sopranos" {
        Sop -- ra -- no words
    }
}
\context Lyrics = "contraltos" {
    \lyricsto "contraltos" {
        Con -- tral -- to words
    }
}
}
>>
}

```



Vous pouvez générer d'autres combinaisons de paroles et portées à partir de ces exemples, ou en examinant les [Section "modèles" dans *Manuel d'initiation*](#) inclus dans le manuel d'initiation.

Morceaux choisis

Espacement des paroles selon les pratiques de la version 2.12

La version 2.14 a donné naissance à un nouveau moteur pour l'espacement vertical des paroles. Celles-ci peuvent donc se retrouver positionnées différemment. Le moteur adoptera les usages de la version 2.12 une fois que vous aurez réglé certaines propriétés des contextes **Lyric** et **Staff**.

```

global = {
    \key d \major
    \time 3/4
}

sopMusic = \relative c' {
    % VERSE ONE
    fis4 fis fis | \break
    fis4. e8 e4
}

altoMusic = \relative c' {
    % VERSE ONE
    d4 d d |
    d4. b8 b4 |
}

tenorMusic = \relative c' {
    a4 a a |
}

```



```

    b4. g8 g4 |
}

bassMusic = \relative c {
    d4 d d |
    g,4. g8 g4 |
}

words = \lyricmode {
    Great is Thy faith- ful- ness,
}

\score {
  \new ChoirStaff <<
    \new Lyrics = sopranos
    \new Staff = women <<
      \new Voice = "sopranos" {
        \voiceOne
        \global \sopMusic
      }
      \new Voice = "altos" {
        \voiceTwo
        \global \altoMusic
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors"
    \new Staff = men <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        \global \tenorMusic
      }
      \new Voice = "basses" {
        \voiceTwo \global \bassMusic
      }
    >>
    \new Lyrics = basses
    \context Lyrics = sopranos \lyricsto sopranos \words
    \context Lyrics = altos \lyricsto altos \words
    \context Lyrics = tenors \lyricsto tenors \words
    \context Lyrics = basses \lyricsto basses \words
  >>
  \layout {
    \context {
      \Lyrics
      \override VerticalAxisGroup #'staff-affinity = ##f
      \override VerticalAxisGroup #'staff-staff-spacing =
        #'((basic-distance . 0)
          (minimum-distance . 2)
          (padding . 2))
    }
  }
}

```

```

\context {
  \Staff
  \override VerticalAxisGroup #'staff-staff-spacing =
    #'((basic-distance . 0)
      (minimum-distance . 2)
      (padding . 2))
}
}
}

```

Great is Thy

Great is Thy

Great is Thy

Great is Thy

faith- ful- ness,

faith- ful- ness,

faith- ful- ness,

faith- ful- ness,

faith- ful- ness,

faith- ful- ness,

Voir aussi

Manuel d'initiation : [Section "Ensemble vocal"](#) dans *Manuel d'initiation*.

Manuel de notation : [Section 5.1.7 \[Ordonnancement des contextes\]](#), page 564, [Section 5.1.2 \[Création d'un contexte\]](#), page 551.

Positionnement horizontal des syllabes

La propriété `#'minimum-distance` de l'objet `Lyrics.LyricSpace` permet d'accroître l'espacement des paroles.

```

{
  c c c c
  \override Lyrics.LyricSpace #'minimum-distance = #1.0
}

```

```

c c c c
}
\addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
}

```



Pour que ce réglage s'applique à toute la partition, définissez-le dans le bloc `\layout`.

```

\score {
  \relative c' {
    c c c c
    c c c c
  }
  \addlyrics {
    longtext longtext longtext longtext
    longtext longtext longtext longtext
  }
  \layout {
    \context {
      \Lyrics
      \override LyricSpace #'minimum-distance = #1.0
    }
  }
}

```



Morceaux choisis

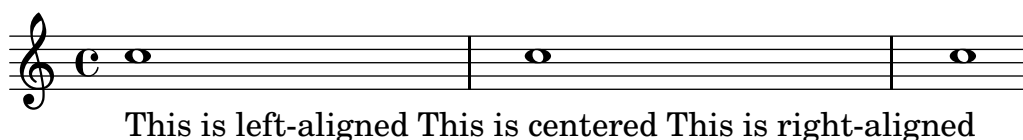
Alignement des syllabes

L'alignement horizontal des paroles peut se gérer à l'aide de la propriété `self-alignment-X` de l'objet `LyricText`. Les valeurs `-1` ou `LEFT` produiront un alignement par la gauche, les valeurs `0` ou `CENTER` un alignement centré, et les valeurs `1` ou `RIGHT` un alignement par la droite.

```

\layout { ragged-right = ##f }
\relative c'' {
  c1
  c1
  c1
}
\addlyrics {
  \once \override LyricText #'self-alignment-X = #LEFT
  "This is left-aligned"
  \once \override LyricText #'self-alignment-X = #CENTER
  "This is centered"
  \once \override LyricText #'self-alignment-X = #1
  "This is right-aligned"
}

```



L'assurance que tous les scripts textuels et les paroles resteront bien à l'intérieur des marges requiert des ressources non négligeables. Afin de réduire le temps de traitement, vous pouvez désactiver cette fonctionnalité en ajoutant

```
\override Score.PaperColumn #'keep-inside-line = ##f
```

Pour s'assurer que les paroles ne seront pas traversées par des barres de mesure, il faut ajouter

```

\layout {
  \context {
    \Lyrics
    \consists "Bar_engraver"
    \consists "Separating_line_group_engraver"
    \override BarLine #'transparent = ##t
  }
}

```

Paroles et reprises

La répétition de *fragments musicaux* est abordée de manière détaillée dans un [Section “chapitre spécifique” dans *Manuel de notation*](#). Nous nous intéresserons ici aux moyens d'ajouter des paroles à des reprises.

Reprises simples

Les paroles attachées à un fragment musical répété devraient adopter rigoureusement la même construction que la musique, si tant est qu'elles ne diffèrent pas d'une fois sur l'autre.

```

\score {
  <<
  \new Staff {
    \new Voice = "melody" {
      \relative c'' {
        a4 a a a
        \repeat volta 2 { b4 b b b }
      }
    }
  }
}

```

```

    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        \repeat volta 2 { Re -- peat -- ed twice. }
      }
    }
  >>
}

```



Les mots seront alors correctement répétés si la reprise est développée.

```

\score {
  \unfoldRepeats {
    <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        \repeat volta 2 { Re -- peat -- ed twice. }
      }
    }
  >>
}

```



Lorsque la reprise est développée et que les paroles diffèrent, saisissez le texte normalement :

```

\score {
  <<
  \new Staff {
    \new Voice = "melody" {
      \relative c'' {
        a4 a a a
        \repeat unfold 2 { b4 b b b }
      }
    }
  }
}

```

```

    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        The first time words.
        Sec -- ond time words.
      }
    }
  >>
}

```



Lorsque les paroles diffèrent pour une reprise non développée – utilisation de *volta* au lieu de *unfold* – les paroles en question doivent être saisies dans des contextes *Lyrics* séparés ; ils seront combinés dans une section parallèle :

```

\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
  \new Lyrics \lyricsto "melody" {
    Not re -- peat -- ed.
    <<
      { The first time words. }
      \new Lyrics {
        \set associatedVoice = "melody"
        Sec -- ond time words.
      }
    >>
  }
  >>
}

```



Et ce quel que soit le nombre de « couplets » :

```

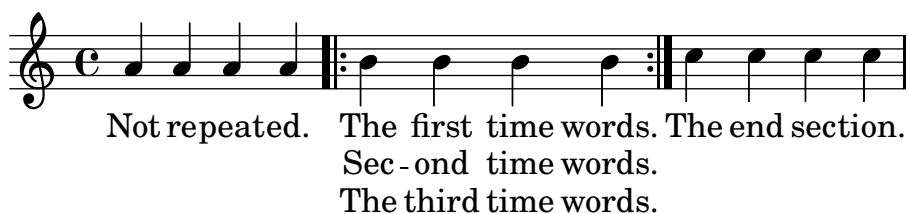
\score {
  <<
    \new Staff {

```

```

\new Voice = "singleVoice" {
  \relative c'' {
    a4 a a a
    \repeat volta 3 { b4 b b b }
    c4 c c c
  }
}
}
\new Lyrics \lyricsto "singleVoice" {
  Not re -- peat -- ed.
  <<
    { The first time words. }
    \new Lyrics {
      \set associatedVoice = "singleVoice"
      Sec -- ond time words.
    }
    \new Lyrics {
      \set associatedVoice = "singleVoice"
      The third time words.
    }
  >>
  The end sec -- tion.
}
>>
}

```



Cependant, lorsque la partition comporte plusieurs portées, cas typique d'un `ChoirStaff`, les paroles des deuxième et troisième couplets seront repoussées sous la dernière portée. L'instruction `alignBelowContext` permet alors de les repositionner correctement :

```

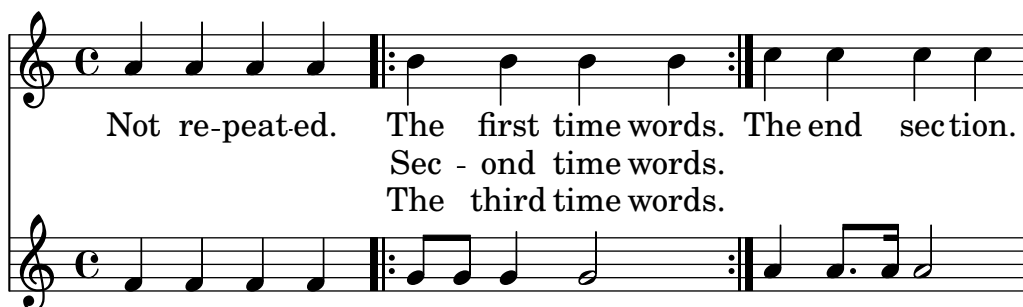
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
  }
  \new Lyrics = "firstVerse" \lyricsto "melody" {
    Not re -- peat -- ed.
    <<
      { The first time words. }
      \new Lyrics = "secondVerse"
    }
  }
}

```

```

\with { alignBelowContext = #"firstVerse" } {
  \set associatedVoice = "melody"
  Sec -- ond time words.
}
\new Lyrics = "thirdVerse"
\with { alignBelowContext = #"secondVerse" } {
  \set associatedVoice = "melody"
  The third time words.
}
>>
The end sec -- tion.
}
\new Voice = "harmony" {
  \relative c' {
    f4 f f f \repeat volta 2 { g8 g g4 g2 } a4 a8. a16 a2
  }
}
>>
}

```



Reprises avec alternative

Les paroles d'un fragment répété, lorsqu'elles sont identiques, peuvent adopter la même construction que la musique :

```

\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 2 { b4 b }
          \alternative { { b b } { b c } }
        }
      }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      Not re -- peat -- ed.
      \repeat volta 2 { Re -- peat -- }
      \alternative { { ed twice. } { ed twice. } }
    }
  }
}

```



```
>>
}
```

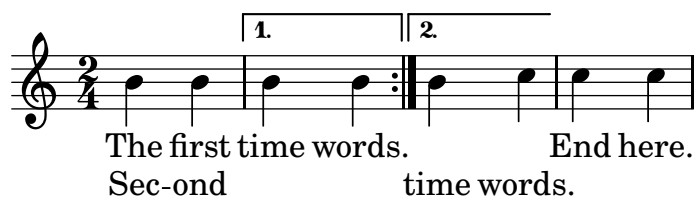


Cette identité de structure n'est toutefois pas possible lorsque les paroles sont différentes. Des instructions `\skip` devront venir s'insérer dans les paroles pour « sauter » les notes des alternatives qui ne les concernent pas.

N'utilisez pas de simple caractère souligné pour sauter une note. N'oubliez pas qu'un caractère souligné indique un mélisme ; la syllabe précédente sera donc alignée à gauche.

Note : La commande `\skip` doit comporter une durée quelle qu'elle soit – elle sera toujours ignorée lorsque les paroles sont associées à une mélodie à l'aide de `\addlyrics` ou `\lyricsto`. Chaque `\skip` correspond à une seule note quelle qu'en soit la durée.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          \repeat volta 2 { b4 b }
          \alternative { { b b } { b c } }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        The first time words.
        \repeat unfold 2 { \skip 1 }
        End here.
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Sec -- ond
        \repeat unfold 2 { \skip 1 }
        time words.
      }
    }
  >>
}
```



Lorsqu'une note se prolonge sur les alternatives, la tenue est indiquée normalement pour la première alternative, et à l'aide de l'instruction `\repeatTie` pour les suivantes. Cette liaison « de répétition » pose problème en matière d'alignement des paroles puisque la longueur de l'alternative est accrue en raison de la liaison.

D'autre part, une liaison de prolongation crée un mélisme qui sera effectif pour la première alternative, mais pas pour les autres. La solution pour « recalculer » les paroles consiste à désactiver temporairement la détection automatique de mélismes et insérer des « blancs ».

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          \set melismaBusyProperties = #'()
          \repeat volta 2 { b4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          \unset melismaBusyProperties
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        \repeat volta 2 { Here's a __ }
        \alternative {
          { \skip 1 verse }
          { \skip 1 sec }
        }
      }
      ond one.
    }
  }
  >>
}
```



Notez bien que l'utilisation conjointe de `\unfoldRepeats` et de `\repeatTie` entraîne l'impression d'une double liaison, sauf à supprimer les `\repeatTie`.

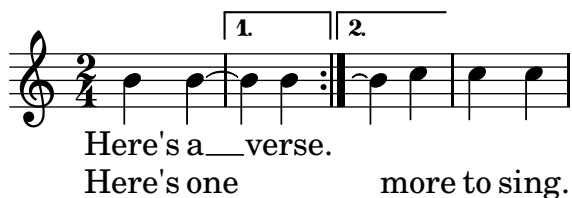
Lorsque les paroles sont différentes sur le fragment répété, la construction avec `\repeat` est inefficace ; vous devrez alors insérer des blancs :

```
\score {
  <<
    \new Staff {
      \time 2/4
```

```

\new Voice = "melody" {
  \relative c'' {
    \repeat volta 2 { b4 b ~}
    \alternative { { b b } { b \repeatTie c } }
    c4 c
  }
}
}
\new Lyrics {
  \lyricsto "melody" {
    Here's a __ verse.
    \repeat unfold 2 { \skip 1 }
  }
}
\new Lyrics {
  \lyricsto "melody" {
    Here's one
    \repeat unfold 2 { \skip 1 }
    more to sing.
  }
}
>>
}

```



Les indications de mélisme et traits d'union en début d'alternative doivent être insérées manuellement :

```

\score {
  <<
  \new Staff {
    \time 2/4
    \new Voice = "melody" {
      \relative c'' {
        \repeat volta 2 { b4 b ~}
        \alternative { { b b } { b \repeatTie c } }
        c4 c
      }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      Here's a __ verse.
      \repeat unfold 2 { \skip 1 }
    }
  }
  \new Lyrics {

```

```

\lyricsto "melody" {
  Here's "a_"
  \skip 1
  "_" sec -- ond one.
}
}
>>
}

```



Voir aussi

Manuel de notation : [Section 5.1.3 \[Conservation d'un contexte\]](#), page 552, [Section 1.4 \[Répétitions et reprises\]](#), page 134.

Paroles alternatives

Il arrive parfois, dans un fragment répété, qu'une note soit divisée pour répondre au texte. Vous pouvez indiquer cette adaptation rythmique en désactivant temporairement la détection automatique des mélismes tout en spécifiant ces mélismes au niveau des paroles :

```

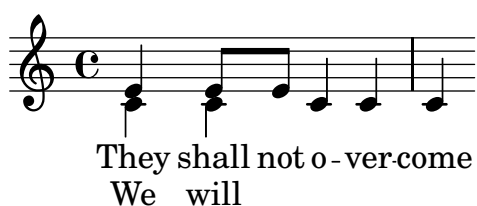
\score {
  <<
    \new Voice = "melody" {
      \relative c' {
        \set melismaBusyProperties = #'()
        \slurDown
        \slurDashed
        e4 e8 ( e ) c4 c |
        \unset melismaBusyProperties
        c
      }
    }
    \new Lyrics \lyricsto "melody" {
      They shall not o -- ver -- come
    }
    \new Lyrics \lyricsto "melody" {
      We will _
    }
  >>
}

```



En donnant un nom à chaque voix et en leur attribuant spécifiquement des paroles, vous pourrez traiter le cas où notes et rythme diffèrent d'une fois sur l'autre :

```
\score {
  <<
    \new Voice = "melody" {
      \relative c' {
        <<
          {
            \voiceOne
            e4 e8 e
          }
          \new Voice = "splitpart" {
            \voiceTwo
            c4 c
          }
        >>
        \oneVoice
        c4 c |
        c
      }
    }
    \new Lyrics \lyricsto "melody" {
      They shall not o -- ver -- come
    }
    \new Lyrics \lyricsto "splitpart" {
      We will
    }
  >>
}
```



2.1.3 Couplets

Numérotation des couplets

On peut ajouter un numéro aux couplets en définissant la variable `stanza` :

```
\new Voice {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set stanza = #"1. "
  Hi, my name is Bert.
} \addlyrics {
  \set stanza = #"2. "
  Oh, ché -- ri, je t'aime
}
```



1. Hi, my name is Bert.
2. Oh, ché - ri, je t'aime

Ces numéros se placeront juste avant le début de la première syllabe.

Indication de nuance dans les couplets

Lorsque des couplets ont des nuances différentes, vous pouvez ajouter une nuance en regard de chacun d'eux. L'objet `StanzaNumber` contient tout ce qui se place avant les paroles du couplet. Pour des raisons techniques, vous devrez définir la variable `stanza` en dehors du mode `\lyricmode`.

```
text = {
  \set stanza = \markup { \dynamic "ff" "1. " }
  \lyricmode {
    Big bang
  }
}
```

```
<<
  \new Voice = "tune" {
    \time 3/4
    g'4 c'2
  }
\new Lyrics \lyricsto "tune" \text
>>
```



ff 1. Big bang

Indication du personnage et couplets

On peut également ajouter le nom de chaque rôle ; ils s'imprimeront au début de chaque ligne comme les noms d'instrument. Il faut pour cela définir `vocalName`, et `shortVocalName` pour une version abrégée.

```
\new Voice {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set vocalName = #"Bert "
  Hi, my name is Bert.
} \addlyrics {
  \set vocalName = #"Ernie "
  Oh, ché -- ri, je t'aime
}
```



Bert Hi, my name is Bert.
Ernie Oh, ché - ri, je t'aime

Rythme différent selon le couplet

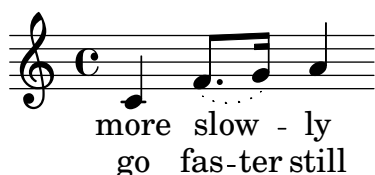
Il arrive assez souvent que les paroles de différents couplets, bien qu'attachées à une même mélodie, ne s'articulent pas de la même manière. La commande `\lyricsto` est cependant capable de gérer de telles situations.

Mélismes dans certaines strophes seulement

Il peut survenir que les paroles comportent un mélisme pour l'un des couplets, mais plusieurs syllabes pour d'autres. Une solution consiste à ignorer temporairement les mélismes dans le couplet ayant le plus de syllabes. Il suffit pour cela de définir la propriété `ignoreMelismata` à l'intérieur du contexte `Lyrics`.

Petit détail qui a son importance : l'activation de `ignoreMelismata` doit **précéder** la syllabe à partir de laquelle elle s'appliquera :

```
<<
\relative c' \new Voice = "lahlah" {
  \set Staff.autoBeaming = ##f
  c4
  \slurDotted
  f8.[( g16)]
  a4
}
\new Lyrics \lyricsto "lahlah" {
  more slow -- ly
}
\new Lyrics \lyricsto "lahlah" {
  go
  \set ignoreMelismata = ##t
  fas -- ter
  \unset ignoreMelismata
  still
}
>>
```



Problèmes connus et avertissements

Contrairement aux autres utilisations de l'instruction `\set`, il n'est pas possible de la faire précéder d'un `\once` dans le cas de `\set ignoreMelismata`. Les paroles affectées par la propriété `ignoreMelismata` **doivent** être encadrées respectivement d'un `\set` et d'un `\unset`.

Syllabe sur note de passage

L'utilisation de la commande `\lyricsto` ne permet pas, par défaut, d'assigner une syllabe à des notes d'ornement – introduites par la commande `\grace`. Vous pouvez cependant y parvenir grâce à la propriété `includeGraceNotes` :

```
<<
\new Voice = melody \relative c' {
  f4 \appoggiatura a32 b4
```

```

\grace { f16[ a16] } b2
\afterGrace b2 { f16[ a16] }
\appoggiatura a32 b4
\acciaccatura a8 b4
}
\new Lyrics
\lyricsto melody {
  normal
  \set includeGraceNotes = ##t
  case,
  gra -- ce case,
  after -- grace case,
  \set ignoreMelismata = ##t
  app. case,
  acc. case.
}
>>

```



Problèmes connus et avertissements

Tout comme pour la propriété `associatedVoice`, la propriété `includeGraceNotes` doit être activée au moins une syllabe avant celle qui viendra s'attacher à la note d'ornement. Dans le cas où cette note se trouve être la première de la pièce, vous devrez recourir à une clause `\with` ou `\context` :

```

<<
\new Voice = melody \relative c' {
  \grace { c16[( d e f] }
  g1) f
}
\new Lyrics \with { includeGraceNotes = ##t }
\lyricsto melody {
  Ah __ fa
}
>>

```



Basculer vers une mélodie alternative

On peut créer des variations plus complexes à partir d'une mélodie à plusieurs voix. Les paroles peuvent suivre l'une ou l'autre des lignes mélodiques, et même basculer de l'une à l'autre si l'on modifie la propriété `associatedVoice`. Dans cet exemple,

```

<<
\relative c' \new Voice = "lahlah" {
  \set Staff.autoBeaming = ##f

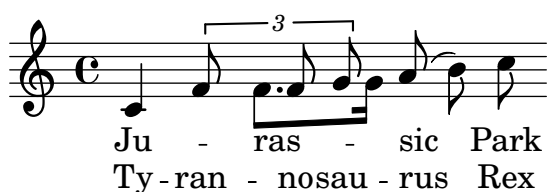
```



```

c4
<<
  \new Voice = "alternative" {
    \voiceOne
    \times 2/3 {
      % show associations clearly.
      \override NoteColumn #'force-hshift = #-3
      f8 f g
    }
  }
  {
    \voiceTwo
    f8.[ g16]
    \oneVoice
  } >>
a8( b) c
}
\new Lyrics \lyricsto "lahlah" {
  Ju -- ras -- sic Park
}
\new Lyrics \lyricsto "lahlah" {
  % Tricky: need to set associatedVoice
  % one syllable too soon!
  \set associatedVoice = alternative % applies to "ran"
  Ty --
  ran --
  no --
  \set associatedVoice = lahlah % applies to "rus"
  sau -- rus Rex
} >>

```



le texte du premier couplet s'aligne de manière habituelle sur la mélodie nommée « lahlah ». Mais le second couplet, tout d'abord rattaché au contexte **lahlah**, bascule sur la mélodie **alternative** pour les syllabes « ran » à « sau » grâce aux lignes

```

\new Lyrics \lyricsto "lahlah" {
  \set associatedVoice = alternative % s'applique à "ran"
  Ty --
  ran --
  no --
  \set associatedVoice = lahlah % s'applique à "rus"
  sau -- rus Rex
}

```

où **alternative** désigne le nom du contexte **Voice** qui contient le triolet.

Notez bien où apparaît la commande `\set associatedVoice` – une syllabe en avance, ce qui est tout à fait correct.

Note : La commande `\set associatedVoice` **doit** intervenir une syllabe *avant* celle qui sera suivie par la bascule. Autrement dit, une modification de la voix associée sera effective une syllabe plus tard que prévu. Il ne s'agit en aucun cas d'un bogue, la raison en est purement technique.

Paroles en fin de partition

Il peut parfois s'avérer opportun d'aligner un seul couplet sur la mélodie et de présenter tous les autres en bloc à la fin du morceau. Ces couplets additionnels peuvent être inclus dans une section `\markup` en dehors du bloc `\score` principal. Vous en trouverez un exemple ci-dessous ; notez également les deux méthodes différentes employées pour indiquer les sauts de ligne, entre les couplets (*verses* en anglais) 2 et 3.

```
melody = \relative c'' {
  \time 2/4
  g4 g8 b | b a b a |
  g4 g8 b | b a b4 |
}

text = \lyricmode {
  \set stanza = #"1." À la clai- re fon- tai- ne,
  M'en al- lant pro- me- ner...
}

\score{ <<
  \new Voice = "one" { \melody }
  \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
}
\markup { \column{
  \line{ Verse 2. }
  \line{ Sous les feuilles d'un chêne }
  \line{ Je me suis fait sécher... }
}
}
\markup{
  \wordwrap-string #"
  Verse 3.

  Chante, rossignol, chante,

  Toi qui as le cœur gai..."
}
```



Verse 2.

Sous les feuilles d'un chêne
Je me suis fait sécher...

Verse 3.

Chante, rossignol, chante,
Toi qui as le cœur gai...

Paroles sur plusieurs colonnes en fin de partition

Si les couplets sont vraiment nombreux, il est possible de les imprimer sur plusieurs colonnes. L'exemple suivant vous montre comment procéder pour que le numéro du couplet soit en retrait à gauche, comme c'est traditionnellement le cas.

```
melody = \relative c'' {
  \time 2/4
  g4 g8 b | b a b a |
  g4 g8 b | b a b4 |
}

text = \lyricmode {
  \set stanza = #"1." À la clai- re fon- tai- ne,
  M'en al- lant pro- me- ner...
}

\score{ <<
  \new Voice = "one" { \melody }
  \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
}

\markup {
  \fill-line {
    \hspace #0.1 % décalage par rapport à la marge de gauche
    % peut être supprimé si l'espace sur la page est réduit
    \column {
      \line { \bold "2."
      \column {
        "Sous les feuilles d'un chêne"
        "Je me suis fait sécher..."
      }
    }
  }
  \vspace #0.1 % ajout d'espace vertical entre les couplets
  \line { \bold "3."
  \column {
    "Chante, rossignol, chante,"
    "Toi qui as le cœur gai..."
  }
}
}
\hspace #0.1 % ajout d'espace horizontal entre les colonnes
\column {
  \line { \bold "4."
  \column {
    "J'ai perdu mon ami"
    "Sans l'avoir mérité..."
  }
}
}
```

```

\vspace #0.1 % ajout d'espace vertical entre les couplets
\line { \bold "5."
  \column {
    "Je voudrais que la rose"
    "Fût encore au rosier..."
  }
}
}
\hspace #0.1 % décalage par rapport à la marge de droite
% peut être supprimé si l'espace sur la page est réduit
}
}

```



1. À la clai-re fon-tai-ne, M'en al-lant pro-me-ner...

2. Sous les feuilles d'un chêne
Je me suis fait sécher...

3. Chante, rossignol, chante,
Toi qui as le cœur gai...

4. J'ai perdu mon ami
Sans l'avoir mérité...

5. Je voudrais que la rose
Fût encore au rosier...

Voir aussi

Référence des propriétés internes : Section “LyricText” dans *Référence des propriétés internes*, Section “StanzaNumber” dans *Référence des propriétés internes*.

2.1.4 Chansons

Références en matière de chanson

Une chanson se présente la plupart du temps sous la forme de trois portées : une pour la mélodie surmontant un système pianistique pour l’accompagnement ; les paroles du premier couplet s’accrochent sous la mélodie. S’il n’y a que deux ou trois couplets, et que cela n’est pas gênant au niveau de l’aspect général, tous peuvent prendre place entre la mélodie et l’accompagnement. Dans le cas contraire, le premier couplet sera imprimé sous la mélodie et les suivants après la partition, sous forme de blocs de texte indépendants.

Tous les éléments qui permettent d’imprimer des chansons sont examinés à différents endroits de la documentation de LilyPond :

- L’agencement des portées est abordé au chapitre Section 1.6.1 [Gravure des portées], page 171.
- Les spécificités du piano sont abordées au chapitre Section 2.2 [Instruments utilisant des portées multiples], page 300.
- L’affectation de paroles à une ligne mélodique est abordée au chapitre Section 2.1.1 [Vue d’ensemble de la musique vocale], page 238.
- Le positionnement des paroles fait l’objet d’une Section “rubrique dédiée” dans *Manuel de notation*.
- La gestion des couplets est abordée dans un Section “chapitre spécifique” dans *Manuel de notation*.

- L'harmonisation d'une chanson est souvent indiquée par des noms d'accord en surplomb de la mélodie. Ceci est abordé au chapitre [Section 2.7.2 \[Gravure des accords\]](#), page 390.
- L'impression de diagrammes d'accord, lorsque l'accompagnement est fait à la guitare, est expliqué à la rubrique « Tablatures sous forme d'étiquette », au chapitre [Section 2.4.1 \[Vue d'ensemble des cordes frettées\]](#), page 317.

Voir aussi

Manuel d'initiation : [Section “Chansons”](#) dans *Manuel d'initiation*.

Manuel de notation : [Section 2.1.1 \[Vue d'ensemble de la musique vocale\]](#), page 238, [Section 2.7.2 \[Gravure des accords\]](#), page 390, [Section 1.6.1 \[Gravure des portées\]](#), page 171, [Section 2.2 \[Instruments utilisant des portées multiples\]](#), page 300, [\[Positionnement vertical des paroles\]](#), page 252, [Section 2.1.3 \[Couplets\]](#), page 267.

Morceaux choisis : [Section “Musique vocale”](#) dans *Morceaux choisis*.

Feuille de chant

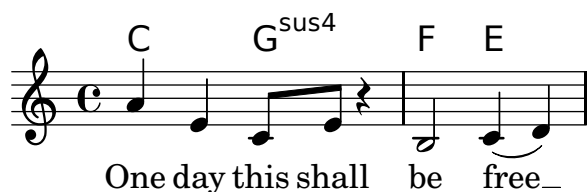
Une simple feuille de chant s'obtient en combinant une partie vocale et son harmonisation. La syntaxe appropriée est expliquée en détails au chapitre [Section 2.7 \[Notation des accords\]](#), page 385.

Morceaux choisis

Chanson simple

Assembler des noms d'accords, une mélodie et des paroles permet d'obtenir la partition d'un chanson :

```
<<
\chords { c2 g:sus4 f e }
\relative c'' {
  a4 e c8 e r4
  b2 c4( d)
}
\addlyrics { One day this shall be free __ }
>>
```



Voir aussi

Manuel de notation : [Section 2.7 \[Notation des accords\]](#), page 385.

2.1.5 Chorale

Nous allons voir, dans les paragraphes qui suivent, les particularités de la musique chorale, qu'il s'agisse de motet, d'oratorio ou de simple partie de chœur.

Références en matière de chorale

Une partition pour chœur comporte habituellement de deux à quatre portées regroupées dans un **ChoirStaff**. L'accompagnement, s'il y en a un, se présente sous la forme d'un système pianistique – un **PianoStaff** – en dessous du chœur ; il s'agira d'une simple réduction dans le

cas d'une œuvre *a capella*. Les notes de chaque pupitre font l'objet d'un contexte **Voice** distinct. Ces contextes **Voice** peuvent se voir groupés ou non sur une même portée.

Les paroles sont traitées dans des contextes **Lyrics** qui viendront se placer tantôt sous la portée, tantôt au-dessus et au-dessous de la portée si elle contient deux voix.

Certaines composantes d'une partition pour chœur sont examinées à différents endroits de la documentation de LilyPond :

- La création pas à pas d'une partition pour chœur se trouve dans le manuel d'initiation, à la rubrique [Section “Partition pour chœur à quatre voix mixtes” dans Manuel d'initiation](#).
- Plusieurs exemples et canevas sont regroupés dans le manuel d'initiation, à la rubrique [Section “Ensemble vocal” dans Manuel d'initiation](#).
- Des informations détaillées sur les contextes **ChoirStaff** et **PianoStaff** sont disponibles au chapitre [\[Regroupement de portées\]](#), page 172.
- Les formes de notation particulière, telle que celle utilisée en *Sacred Harp* et assimilées, sont abordées au chapitre [\[Têtes de note à forme variable\]](#), page 36.
- Lorsque plusieurs pupitres sont regroupés sur la même portée, les hampes, liaisons, etc. de la voix supérieure sont orientées vers le haut, et inversement pour la voix inférieure. L'utilisation de `\voiceOne` et `\voiceTwo` est expliquée au chapitre [\[Polyphonie sur une portée\]](#), page 155.

Commandes prédéfinies

`\oneVoice`, `\voiceOne`, `\voiceTwo`.

Voir aussi

Manuel d'initiation : [Section “Partition pour chœur à quatre voix mixtes” dans Manuel d'initiation](#), [Section “Ensemble vocal” dans Manuel d'initiation](#).

Manuel de notation : [Section 5.1.7 \[Ordonnancement des contextes\]](#), page 564, [\[Regroupement de portées\]](#), page 172, [\[Têtes de note à forme variable\]](#), page 36, [\[Polyphonie sur une portée\]](#), page 155.

Morceaux choisis : [Section “Musique vocale” dans Morceaux choisis](#).

Référence des propriétés internes : [Section “ChoirStaff” dans Référence des propriétés internes](#), [Section “Lyrics” dans Référence des propriétés internes](#), [Section “PianoStaff” dans Référence des propriétés internes](#).

Mise en forme d'une partition chorale

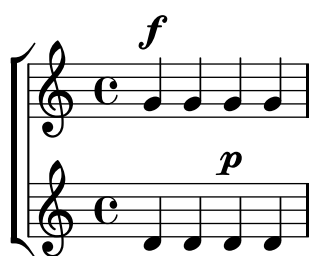
Une partition pour chœur sur quatre portées, avec ou sans accompagnement, présente traditionnellement deux systèmes par page. Selon la taille du papier, vous pourrez être amené à effectuer quelques ajustements aux réglages par défaut, notamment en raison des points suivants :

- La taille des portées a des répercussions sur l'ensemble des éléments de la partition. Voir [Section 4.2.2 \[Définition de la taille de portée\]](#), page 509.
- La distance séparant les systèmes, les portées et les paroles peut s'ajuster de manière séparée, comme expliqué au chapitre [Section 4.4 \[Espace vertical\]](#), page 518.
- La mise en évidence des différentes dimensions permet d'appréhender avec finesse le réglage des variables d'espacement vertical et, pourquoi pas, de faire tenir la partition sur moins de pages, comme l'explique la rubrique [Section 4.6 \[Réduction du nombre de pages de la partition\]](#), page 545.
- Lorsque le nombre de systèmes varie d'une page à l'autre, il est judicieux de l'indiquer visuellement, en suivant les instructions de la rubrique [\[Séparation des systèmes\]](#), page 177.

- Pour de plus amples détails quant aux propriétés liées au formatage, consultez le chapitre [Section 4.1 \[Mise en forme de la page\]](#), page 496.

Les indications de nuance se placent traditionnellement sous la portée, ce qui n'est pas le cas en matière de musique vocale dans le but d'éviter toute collision avec les paroles. La commande prédéfinie `\dynamicUp` attachée à un contexte `Voice` permet de positionner les nuances au-dessus de la portée. Dans le cas où il y en aurait plusieurs, cette commande devra apparaître dans chacun des contextes `Voice` qui le requiert. Vous pouvez aussi opter pour la forme développée, comme dans l'exemple ci-dessous, pour que cela s'applique à toutes les portées de la partition – changez `\Score` en `\ChoirStaff` s'il y a d'autres parties que celles du chœur.

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice {
        \relative c'' { g4\f g g g }
      }
    }
    \new Staff {
      \new Voice {
        \relative c' { d4 d d\p d }
      }
    }
  >>
  \layout {
    \context {
      \Score
      \override DynamicText #'direction = #UP
      \override DynamicLineSpanner #'direction = #UP
    }
  }
}
```



Commandes prédéfinies

`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`.

Voir aussi

Manuel de notation : [Section 4.6.2 \[Modification de l'espacement\]](#), page 547, [Section 4.6.1 \[Mise en évidence de l'espacement\]](#), page 546, [Section 4.6 \[Réduction du nombre de pages de la partition\]](#), page 545, [Section 4.1 \[Mise en forme de la page\]](#), page 496, [Section 4.2 \[Mise en forme de la partition\]](#), page 507, [\[Séparation des systèmes\]](#), page 177, [Section 4.2.2 \[Définition de la taille de portée\]](#), page 509, [Section 4.3.8 \[Recours à une voix supplémentaire pour gérer les sauts\]](#), page 516, [Section 4.4 \[Espacement vertical\]](#), page 518.

Référence des propriétés internes : [Section “VerticalAxisGroup”](#) dans *Référence des propriétés internes*, [Section “StaffGrouper”](#) dans *Référence des propriétés internes*.

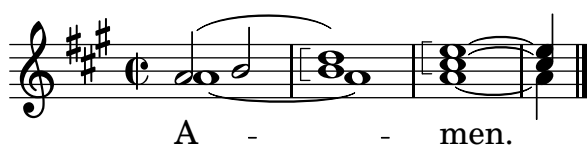
Division de voix

Utilisation d'un `arpeggioBracket` pour rendre les divisions plus évidentes

Un crochet d'arpège (`arpeggioBracket`) permet de mettre en évidence les divisions d'un pupitre en l'absence de hampe, comme on le voit régulièrement dans les partitions pour chœur.

```
\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper"
      <<
        { \voiceOne \arpeggioBracket
          a2( b2
            <b d>1\arpeggio)
            <cs e>\arpeggio ~
            <cs e>4
          }
          \addlyrics { \lyricmode { A -- men. } }
        >>
      \new Voice = "lower"
      { \voiceTwo
        a1 ~
        a
        a ~
        a4 \bar "|"
      }
    >>
  }
  \layout { ragged-right = ##t }
}
```



Voir aussi

Manuel de notation : [Section 1.3.3 \[Signes d'interprétation sous forme de ligne\]](#), page 125.

2.1.6 Opéras et musiques de scène

Tout ce qui permet d'exécuter un opéra ou une œuvre scénique accompagnée de musique se présente généralement sous l'une ou plusieurs des formes suivantes :

- Un *Conducteur* destiné au chef d'orchestre. Il comprend l'intégralité des parties d'orchestre et des chanteurs, ainsi que des citations du livret pour les passages déclamés.
- Un *matériel d'orchestre*, autrement dit une partition pour chacun des pupitres de l'orchestre ou de l'ensemble instrumental.
- Une *partition de chœur* regroupant toutes les parties vocales avec accompagnement au piano. Cet accompagnement est souvent une réduction d'orchestre où les différents instruments sont

indiqués. Les partitions de chœur comprennent parfois des indications de mise en scène ainsi que des extraits du livret.

- Une *partition de choriste* qui ne comprend que les parties vocales – donc sans accompagnement. Elle peut être augmentée du livret.
- Un *livret* contenant l'intégralité des dialogues et le texte des passages chantés. On y trouve aussi très souvent les indications de mise en scène. Bien que LilyPond soit capable de « typographier » un livret, n'oubliez pas qu'il n'y a dans ce cas pas de musique, et que d'autres outils pourraient être mieux appropriés.

La plupart de ce qui est nécessaire à la mise en forme d'un opéra ou d'une musique de scène est disséminé dans la somme documentaire de LilyPond. Nous commencerons par rappeler ces différents éléments, avant que d'en examiner certaines particularités adaptées aux styles opératique et scénique.

Références en matière d'opéra et musique de scène

- Un conducteur contient un certain nombre de portées et de nombreuses paroles. Les manières d'agencer les portées sont indiquées à la rubrique [Regroupement de portées], page 172, et les façons de les combiner à la rubrique [Imbrication de regroupements de portées], page 176.
- Les portées vides sont la plupart du temps éliminées d'un conducteur ou d'une partition de chœur. La réalisation d'une telle partition – les anglophones la disent « à la française » – est expliquée à la rubrique [Masquage de portées], page 185.
- La génération d'un matériel d'orchestre fait l'objet de la rubrique Section 1.6.3 [Écriture de parties séparées], page 188. D'autres parties du chapitre consacré à la notation spécialisée vous seront utiles selon l'orchestration de la pièce. Tous les instruments ne sont pas accordés pareil ; vous trouverez des informations à ce sujet à la rubrique [Instruments transpositeurs], page 22.
- Lorsque le nombre de systèmes varie d'une page à l'autre, il peut être judicieux de les mettre en évidence, en suivant les indications de la rubrique [Séparation des systèmes], page 177.
- Les différentes propriétés impliquées dans la mise en page sont répertoriées au chapitre Section 4.1 [Mise en forme de la page], page 496.
- L'insertion de dialogues et d'indications de mise en scène peuvent se réaliser à l'aide de *markups*, en suivant les directives fournies aux chapitres Section 3.2.3 [Notes de bas de page], page 463 et Section 1.8 [Texte], page 212. Les indications de mise en scène peuvent s'insérer entre deux blocs `\score` selon les préceptes de la rubrique [Texte indépendant], page 219.

Voir aussi

Glossaire musicologique : Section “Partition à la française” dans *Glossaire*, Section “Frenched staves” dans *Glossaire*, Section “instrument transpositeur” dans *Glossaire*.

Manuel de notation : Section 1.8.1 [Ajout de texte], page 212, [Imbrication de regroupements de portées], page 176, [Instruments transpositeurs], page 22, [Masquage de portées], page 185, Section 4.1 [Mise en forme de la page], page 496, Section 3.2.3 [Notes de bas de page], page 463, [Regroupement de portées], page 172, [Séparation des systèmes], page 177, [Transposition], page 10, Section 1.6.3 [Écriture de parties séparées], page 188.

Morceaux choisis : Section “Musique vocale” dans *Morceaux choisis*.

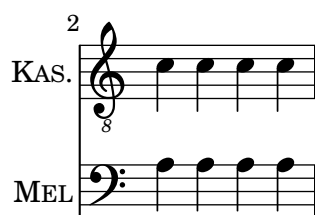
Indication du rôle

Lorsqu'un rôle est distribué sur une portée spécifique, vous pouvez l'indiquer en regard de cette portée :

```

\score {
  <<
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Kaspar
      \set Staff.shortVocalName = \markup \smallCaps Kas.
      \relative c' {
        \clef "G_8"
        c4 c c c
        \break
        c4 c c c
      }
    }
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Melchior
      \set Staff.shortVocalName = \markup \smallCaps Mel
      \clef "bass"
      \relative c' {
        a4 a a a
        a4 a a a
      }
    }
  >>
}

```



Lorsque la même portée sert à plusieurs personnages, leur nom est généralement imprimé en surplomb de la portée, à chaque changement de rôle. L'utilisation d'un *markup* – dans une fonte réservée à cet effet – vous permettra de générer ces indications :

```

\clef "G_8"
c4^\markup \fontsize #1 \smallCaps Kaspar
c c c
\clef "bass"
a4^\markup \fontsize #1 \smallCaps Melchior
a a a
\clef "G_8"
c4^\markup \fontsize #1 \smallCaps Kaspar
c c c

```



Dans le cas où les changements de personnage se multiplient, il peut s'avérer pratique de détourner l'utilisation de la fonction `\instrumentSwitch` pour gérer les différentes interventions de l'un ou de l'autre ; vous devrez auparavant avoir établi la définition d'un « instrument » pour chacun des rôles en question :

```
\addInstrumentDefinition #"kaspar"
#`((instrumentTransposition . ,(ly:make-pitch -1 0 0))
  (shortInstrumentName . "Kas.")
  (clefGlyph . "clefs.G")
  (clefOctavation . -7)
  (middleCPosition . 1)
  (clefPosition . -2)
  (instrumentCueName . ,(markup #:fontsize 1 #:smallCaps "Kaspar"))
  (midiInstrument . "voice oohs"))

\addInstrumentDefinition #"melchior"
#`((instrumentTransposition . ,(ly:make-pitch 0 0 0))
  (shortInstrumentName . "Mel.")
  (clefGlyph . "clefs.F")
  (clefOctavation . 0)
  (middleCPosition . 6)
  (clefPosition . 2)
  (instrumentCueName . ,(markup #:fontsize 1 #:smallCaps "Melchior"))
  (midiInstrument . "voice aahs"))

\relative c' {
  \instrumentSwitch "kaspar"
  c4 c c c
  \instrumentSwitch "melchior"
  a4 a a a
  \instrumentSwitch "kaspar"
  c4 c c c
}
```



Voir aussi

Manuel de notation : [Noms d'instrument], page 188, Section A.21 [Fonctions Scheme], page 732, Section 1.8 [Texte], page 212, [\[Text markup commands\]](#), page [\[undefined\]](#).

Manuel d'extension des fonctionnalités : Section "Construction d'un markup en Scheme" dans *Extension de LilyPond*.

Citation-repère

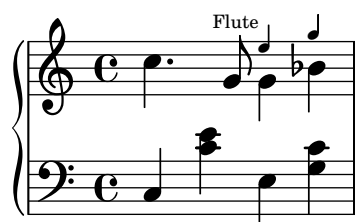
Les citations d'instruments insérées dans les parties vocales, les partitions de chœur ou les partitions d'un pupitre permettent d'indiquer ce qui se passe ailleurs juste avant une entrée. On les retrouve souvent dans la réduction pour piano, ce qui fournit au chef de chœur de précieuses indications sur qui joue quoi, lorsqu'il ne dispose pas d'un conducteur en bonne et due forme.

Les mécanismes de base permettant d'insérer des citations sont expliqués en détail aux rubriques [Citation d'autres voix], page 192 et [Mise en forme d'une citation], page 195. Dans le cas où les citations concernent différents instruments, faire mention de celui qui intervient devient une nécessité ; voici une illustration de la manière de procéder en pareil cas :

```
flute = \relative c'' {
  s4 s4 e g
}
\addQuote "flute" { \flute }

pianoRH = \relative c'' {
  c4. g8
  % position name of cue-ing instrument just before the cue notes,
  % and above the staff
  \new CueVoice {
    \override InstrumentSwitch #'self-alignment-X = #RIGHT
    \set instrumentCueName = "Flute"
  }
  \cueDuring "flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  \new PianoStaff <<
    \new Staff {
      \pianoRH
    }
    \new Staff {
      \clef "bass"
      \pianoLH
    }
  >>
}
```



La citation peut concerner un instrument transpositeur, auquel cas il faudra mentionner sa tonalité dans sa définition, afin que ses hauteurs soient automatiquement converties dans la réplique. Ceci est illustré par l'exemple ci-dessous, dans lequel il est fait appel à une clarinette en si bémol. Dans la mesure où les notes citées se trouvent vers le bas de la portée, nous affectons un DOWN à la commande `\cueDuring`, de telle sorte que les hampes aillent vers le bas et que le nom de l'instrument cité soit en dessous de la portée. Vous noterez que la voix contenant la main droite du piano est explicitement déclarée ; ceci tient au fait que la citation se produit dès le début de la première mesure – si nous ne le faisons pas, l'intégralité de la main droite se verrait placée dans un contexte `CueVoice`.

```
clarinet = \relative c' {
  \transposition bes
```

```

    fis4 d d c
  }
  \addQuote "clarinet" { \clarinet }

pianoRH = \relative c'' {
  \transposition c'
  % position name of cue-ing instrument below the staff
  \new CueVoice {
    \override InstrumentSwitch #'self-alignment-X = #RIGHT
    \override InstrumentSwitch #'direction = #DOWN
    \set instrumentCueName = "Clar."
  }
  \cueDuring "clarinet" #DOWN { c4. g8 }
  g4 bes4
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}

```



Il est clair, au vu de ces deux exemples, que multiplier le nombre de citations dans une partition vocale demande un travail fastidieux et que relire la partie de piano deviendra vite un cauchemar. Vous pouvez néanmoins, comme l'illustre l'exemple suivant, définir une fonction musicale dans le but de vous épargner de la saisie tout en améliorant la lisibilité des notes du piano.

Morceaux choisis

Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur

Lorsque le nombre d'instruments cités dans la réduction pour piano se multiplie, vous pourriez avoir intérêt à créer votre propre fonction pour gérer ces repères. La fonction musicale `\cueWhile` prend quatre arguments : la musique d'où provient la citation, telle que définie par `\addQuote`,

le nom qui sera mentionné en regard de cette citation, son positionnement – UP ou DOWN selon qu'il sera attribué à \voiceOne et placé au-dessus ou \voiceTwo et placé en dessous – et enfin la musique du piano qui interviendra en parallèle. Le nom de l'instrument en question viendra s'aligner sur la gauche de la citation. Bien que vous puissiez effectuer plusieurs citations, elle ne peuvent être simultanées.

```
cueWhile =
#(define-music-function
  (parser location instrument name dir music)
  (string? string? ly:dir? ly:music?)
  #{
    \cueDuring $instrument #dir {
      \once \override TextScript #'self-alignment-X = #RIGHT
      \once \override TextScript #'direction = $dir
      <>-\markup { \tiny #name }
      $music
    }
  })

flute = \relative c'' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }

clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
  }
}
```

```

\new PianoStaff <<
  \new Staff {
    \new Voice {
      \pianoRH
    }
  }
  \new Staff {
    \clef "bass"
    \pianoLH
  }
>>
>>
}

```



Voir aussi

Glossaire musicologique : [Section “cue-notes”](#) dans *Glossaire*.

Manuel de notation : [Section 5.5.1 \[Alignement des objets\]](#), page 593, [Section 5.4.2 \[Direction et positionnement\]](#), page 579, [\[Mise en forme d’une citation\]](#), page 195, [\[Citation d’autres voix\]](#), page 192, [Section 5.6 \[Utilisation de fonctions musicales\]](#), page 602.

Morceaux choisis : [Section “Musique vocale”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “InstrumentSwitch”](#) dans *Référence des propriétés internes*, [Section “CueVoice”](#) dans *Référence des propriétés internes*.

Problèmes connus et avertissements

`\cueDuring` crée automatiquement un contexte `CueVoice` qui accueillera toutes les notes répliquées. Il est par conséquent impossible de faire se superposer des citations à l’aide de la technique simplifiée telle que nous venons de le voir. La superposition de fragments cités requiert que les contextes `CueVoice` soient explicitement déclarés, ainsi que l’utilisation de la commande `\quoteDuring` pour extraire et insérer les notes répliquées.

Musique parlée

Le *parlato* – ou *Sprechgesang* pour les germanistes – est du texte scandé en rythme, mais sans hauteurs définies ; il est indiqué par des notes en croix, à l’instar des percussions – voir [\[Têtes de note spécifiques\]](#), page 33.

Dialogue et musique

Les dialogues que l’on ajoute à la musique sont traditionnellement imprimés en italique au-dessus des portées, au moment même où ils surviennent.

Une courte intervention peut se formuler à l'aide d'un simple *markup* :

```
a4^\markup { \smallCaps { Alex - } \italic { He's gone } } a a a
a4 a a^\markup { \smallCaps { Bethan - } \italic Where? } a
a4 a a a
```



Une intervention un peu plus longue peut nécessiter d'étirer la musique de telle sorte que le texte ait suffisamment de place. LilyPond ne disposant d'aucun mécanisme permettant d'automatiser l'étirement, vous devrez probablement ajuster vous même la mise en forme.

Dans le cas d'une phrase entière ou de passages relativement denses, le recours à un contexte `Lyrics` peut donner de meilleurs résultats. Le contexte `Lyrics` en question ne doit être rattaché à aucune voix musicale ; chaque fragment de dialogue devra donc comporter des durées explicites. Lorsque les dialogues comportent des pauses, le dernier mot devra être séparé du reste et les durées individualisées pour obtenir un espacement harmonieux de la musique.

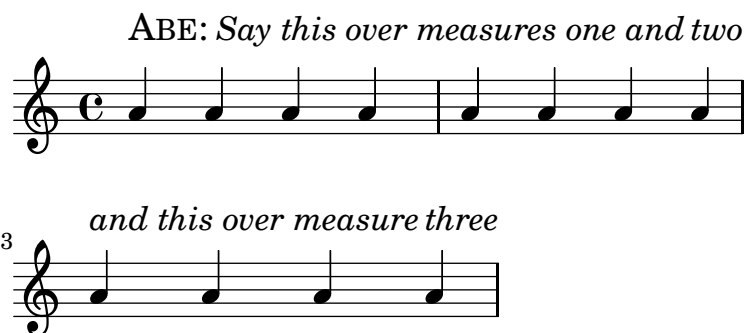
Des dialogues qui s'étendent sur plus d'une ligne vous obligeront à insérer des `\break` et ajuster leur placement pour éviter qu'ils ne débordent dans la marge droite. Le dernier mot de la dernière mesure d'une ligne doit être saisi sur une ligne à part.

Voici une illustration de tout ce que nous venons de voir :

```
music = \relative c'' {
  \repeat unfold 3 { a4 a a a }
}

dialogue = \lyricmode {
  \markup {
    \fontsize #1 \upright \smallCaps Abe:
    "Say this over measures one and"
  }4*7
  "two"4 |
  \break
  "and this over measure"4*3
  "three"4 |
}

\score {
  <<
    \new Lyrics \with {
      \override LyricText #'font-shape = #'italic
      \override LyricText #'self-alignment-X = #LEFT
    }
    { \dialogue }
    \new Staff {
      \new Voice { \music }
    }
  >>
}
```

Voir aussi

Manuel de notation : [\[Durée explicite des syllabes\]](#), page 245, Section 1.8 [\[Texte\]](#), page 212.

Référence des propriétés internes : [Section “LyricText”](#) dans *Référence des propriétés internes*.

2.1.7 Chants liturgiques

Selon les chapelles, la mise en forme des cantiques, psaumes et hymnes répond à des canons bien établis. Bien que différents de par leur présentation, nous verrons dans ce qui suit que les problèmes qui surviennent en matière de typographie se ressemblent, quelle que soit l’obédience.

Références en matière de chant liturgique

La présentation du plain chant et du grégorien selon différents styles est abordée au chapitre [Section 2.9 \[Notations anciennes\]](#), page 407.

Voir aussi

Manuel de notation : [Section 2.9 \[Notations anciennes\]](#), page 407.

Morceaux choisis : [Section “Musique vocale”](#) dans *Morceaux choisis*.

Cantiques et hymnes

La mise en forme contemporaine de cantiques utilise à la fois la notation moderne et un certain nombre d’éléments propres aux notations anciennes. Nous allons examiner quelques uns de ces éléments et la méthode consacrée pour les mettre en œuvre.

Les cantiques utilisent souvent des noires dépourvues de hampe pour indiquer les hauteurs ; le rythme de la mélodie est donné par le rythme et l’accentuation des paroles elles-mêmes.

```
stemOff = { \override Staff.Stem #'transparent = ##t }
```

```
\relative c' {
  \stemOff
  a'4 b c2 |
}
```



Les barres de mesure sont absentes dans la plupart des cas ; celles que vous rencontrerez seront raccourcies ou en pointillé, dans le but d’indiquer une « respiration ». Le fait de supprimer le graveur de barres de mesure produira des portées sans barre :

```
\score {
  \new StaffGroup <<
    \new Staff {
      \relative c'' {
```

```

        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
    }
}
\new Staff {
    \relative c'' {
        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
    }
}
>>
\layout {
    \context {
        \Staff
        \remove "Bar_engraver"
    }
}
}

```



L'absence de barre de mesure peut ne concerner que certaines portées :

```

\score {
    \new ChoirStaff <<
        \new Staff
        \with { \remove "Bar_engraver" } {
            \relative c'' {
                a4 b c2 |
                a4 b c2 |
                a4 b c2 |
            }
        }
    \new Staff {
        \relative c'' {
            a4 b c2 |
            a4 b c2 |
            a4 b c2 |
        }
    }
}
>>
}

```



L'absence de barre de mesure sur un fragment seulement s'obtient en traitant ce fragment comme une cadence. S'il est relativement long, pensez à y insérer des barres fantômes – un simple `\bar ""` – pour indiquer à LilyPond où serait susceptible de se produire un saut de ligne.

```
a4 b c2 |
\cadenzaOn
a4 b c2
a4 b c2
\bar ""
a4 b c2
a4 b c2
\cadenzaOff
a4 b c2 |
a4 b c2 |
```



Dans la mélodie d'un cantique, les silences ou pauses s'indiquent à l'aide de barres de mesure spécifiques :

```
a4
\cadenzaOn
b c2
a4 b c2
\bar ""
a4 b c2
a4 b c2
\bar ":"
a4 b c2
\bar "dashed"
a4 b c2
\bar "||"
```



Vous pouvez accessoirement, bien qu'il s'agisse de notation moderne, emprunter au grégorien des indications de pause et silence. Il vous suffit pour cela d'adapter la commande `\breathe` selon vos besoins :

```
divisioMinima = {
  \once \override BreathingSign #'stencil = #ly:breathing-sign::divisio-minima
  \once \override BreathingSign #'Y-offset = #0
  \breathe
}
divisioMaior = {
  \once \override BreathingSign #'stencil = #ly:breathing-sign::divisio-maior
```

```

\once \override BreathingSign #'Y-offset = #0
\breathes
}
divisioMaxima = {
  \once \override BreathingSign #'stencil = #ly:breathing-sign::divisio-maxima
  \once \override BreathingSign #'Y-offset = #0
  \breathes
}
finalis = {
  \once \override BreathingSign #'stencil = #ly:breathing-sign::finalis
  \once \override BreathingSign #'Y-offset = #0
  \breathes
}

\score {
  \relative c'' {
    g2 a4 g
    \divisioMinima
    g2 a4 g
    \divisioMaior
    g2 a4 g
    \divisioMaxima
    g2 a4 g
    \finalis
  }
  \layout {
    \context {
      \Staff
      \remove "Bar_engraver"
    }
  }
}

```



De nombreux cantiques sont dépourvus de métrique, voire même de clef.

```

\score {
  \new Staff {
    \relative c'' {
      a4 b c2 |
      a4 b c2 |
      a4 b c2 |
    }
  }
  \layout {
    \context {
      \Staff
      \remove "Bar_engraver"
      \remove "Time_signature_engraver"
      \remove "Clef_engraver"
    }
  }
}

```

```

    }
  }
}

```



L'une des traditions anglicanes est de chanter les psaumes sur la base d'un fragment de sept mesures – forme *single* ou simple – ou de deux fragments toujours de sept mesures – forme *double*. Chaque fragment est divisé en deux parties correspondant aux deux moitiés de chaque verset et généralement séparées par une double barre. Il n'est fait usage que de rondes et de blanches, et la première mesure de chaque moitié contient un simple accord de rondes. Il s'agit donc des notes correspondant au « récitatif ». Ces cantiques sont traditionnellement centrés sur la page.

```

SopranoMusic = \relative g' {
  g1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

AltoMusic = \relative c' {
  e1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

TenorMusic = \relative a {
  c1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

BassMusic = \relative c {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
}

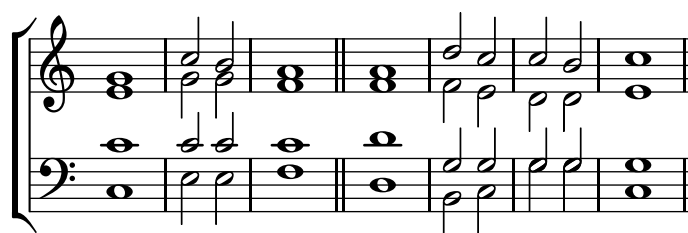
% Use markup to center the chant on the page
\markup {
  \fill-line {
    \score { % centered
      <<
        \new ChoirStaff <<
          \new Staff <<
            \global
            \clef "treble"
            \new Voice = "Soprano" <<
              \voiceOne
              \SopranoMusic
            >>
          >>
        >>
      >>
    }
  }
}

```

```

\new Voice = "Alto" <<
  \voiceTwo
  \AltoMusic
>>
>>
\new Staff <<
  \clef "bass"
  \global
  \new Voice = "Tenor" <<
    \voiceOne
    \TenorMusic
  >>
  \new Voice = "Bass" <<
    \voiceTwo
    \BassMusic
  >>
>>
>>
>>
\layout {
  \context {
    \Score
    \override SpacingSpanner
      #'base-shortest-duration = #(ly:make-moment 1 2)
  }
  \context {
    \Staff
    \remove "Time_signature_engraver"
  }
}
} % End score
} % End markup

```



D'autres approches d'une telle mise en forme font l'objet du premier des exemples qui suivent.

Morceaux choisis

Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

```

stemOn = {
  \revert Staff.Stem #'transparent
  \revert Staff.Flag #'transparent
}

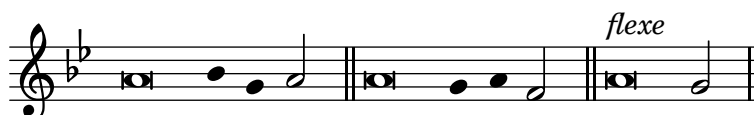
```

```

stemOff = {
  \override Staff.Stem #'transparent = ##t
  \override Staff.Flag #'transparent = ##t
}

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \bar "||"
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \bar "||"
    \stemOff a'\breve^{\markup { \italic flexe }}
    \stemOn g'2 \bar "||"
  }
}

```



Cantiques et autres textes liturgiques peuvent être mis en forme avec une grande liberté et parfois emprunter des éléments de notation ancienne. Le texte apparaît souvent sous la mélodie, les mots alors alignés sur les notes. En pareil cas, les notes sont espacées selon les syllabes et non leur durée.

Exemples de notation ancienne – transcription moderne de musique grégorienne

Voici comment vous pourriez transcrire du grégorien. Pour mémoire, il n'y a en grégorien ni de découpage en mesure, ni de hampe ; seules sont utilisées des têtes de note blanches ou noires, ainsi que des signes spécifiques permettant d'indiquer des silences de différentes durées.

```

\include "gregorian.ly"

chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g) a2 \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {

```

```

\context {
  \Staff
  \remove "Time_signature_engraver"
  \remove "Bar_engraver"
  \override Stem #'transparent = ##t
  \override Flag #'transparent = ##t
}
\context {
  \Voice
  \override Stem #'length = #0
}
\context {
  \Score
  barAlways = ##t
}
}
}

```



Voir aussi

Manuel d’initiation : Section “Visibilité et couleur des objets” dans *Manuel d’initiation*, Section “Ensemble vocal” dans *Manuel d’initiation*.

Manuel de notation : Section 2.9 [Notations anciennes], page 407, [Barres de mesure], page 90, Section 5.1.4 [Modification des greffons de contexte], page 555, Section 2.9.3 [Typographie du chant grégorien], page 419, [Musique sans métrique], page 67, Section 5.4.6 [Visibilité des objets], page 586.

Psalmodie

Les versets d’un psaume anglican sont habituellement centrées sous la mélodie.

Dans le cas d’un chant simple, les sept mesures qui le composent sont répétées pour chaque verset. Dans le cas d’un chant double, les quatorze mesures se répètent par couple de versets. Des marques insérées dans le texte indiquent comment il s’articule par rapport à la mélodie. Chaque verset est séparé en deux, et la rupture est indiquée par un caractère deux points (:) correspondant à la double barre de la mélodie. Le texte précédant les deux points se chante sur les trois premières mesures, celui qui suit sur les quatre dernières mesures.

De simples barres verticales – remplacées par des virgules inversées dans certains psautiers – représentent les barres de mesures portées sur la mélodie. En mode *markup*, ces barres s’obtiennent en saisissant le même caractère | qui sert pour les contrôles de mesure.

```

\markup {
  \fill-line {
    \column {
      \left-align {
        \line { 0 come let us sing | unto the | Lord : let }
        \line { us heartily rejoice in the | strength of | our }
        \line { sal- | -vation. }
      }
    }
  }
}

```



```

    }
  }
}

```

O come let us sing | unto the | Lord : let
us heartily rejoice in the | strength of | our
sal- | -vation.

Vous pourriez tout à fait utiliser d'autres symboles disponibles au travers des glyphes de la fonte **fetaMusic** – voir le chapitre [Section 1.8.3 \[Fontes\]](#), page 234 pour plus de détails.

```

tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { 0 come let us sing \tick unto the \tick Lord : let }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
      }
      \line { sal \tick vation. }
    }
  }
}
}

```

O come let us sing' unto the' Lord : let
us heartily rejoice in the' strength of' our
sal' vation.

Lorsqu'une mesure ne comporte qu'une ronde, le texte correspondant à cette mesure est chanté sur cette même note, selon le rythme naturel de la phrase. Lorsque la mesure comporte deux notes, celles-ci correspondent en général à une ou deux syllabes ; dans le cas contraire, le changement de note est indiqué par un point.

```

dot = \markup {
  \raise #0.7 \musicglyph #"dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          0 come let us sing \tick unto \dot the \tick Lord : let
        }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
      }
    }
  }
}

```

```

        \line { sal \tick vation. }
    }
}
}

```

O come let us sing' unto • the' Lord : let
us heartily rejoice in the' strength of' our
sal' vation.

Certains psautiers font apparaître, pour indiquer une césure, une astérisque au lieu d'une virgule, ainsi que des caractères gras pour les syllabes accentuées ou allongées.

```

dot = \markup {
  \raise #0.7 \musicglyph #"dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { Today if ye will hear his voice * }
        \line {
          \concat { \bold hard en }
          | not your | hearts : as in the pro-
        }
        \line { vocation * and as in the \bold day of tempt- | }
        \line { -ation | in the | wilderness. }
      }
    }
  }
}

```

Today if ye will hear his voice *
harden | not your | hearts : as in the pro-
vocation * and as in the **day** of tempt- |
-ation | in the | wilderness.

D'autres psautiers indiquent une syllabe accentuée en la surchargeant d'un accent.

```

tick = \markup {
  \raise #2 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          O come let us \concat {
            si \combine \tick ng
          }
        }
      }
    }
  }
}

```

```

        | unto the | Lord : let
    }
    \line {
        us heartily \concat {
            rejo \combine \tick ice
        }
        in the | strength of | our
    }
    \line { sal- | -vation. }
}
}
}
}

```

O come let us [´]sing | unto the | Lord : let
us heartily rejo[´]ice in the | strength of | our
sal- | -vation.

L'utilisation du mode *markup* pour centrer le texte et agencer les lignes est abordée en détails au chapitre [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

La plupart de ces éléments sont regroupés dans l'un des versets du modèle « Psalmodie », à la rubrique [Section “Ensemble vocal”](#) dans *Manuel d'initiation*.

Voir aussi

Manuel d'initiation : [Section “Ensemble vocal”](#) dans *Manuel d'initiation*.

Manuel de notation : [Section 1.8.3 \[Fontes\]](#), page 234, [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

Mesure incomplète et musique liturgique

Il arrive fréquemment que les chants liturgiques comportent des mesures incomplètes aussi bien en début qu'en fin de ligne, de telle sorte qu'à une portée corresponde une ligne de texte. Ceci requiert donc l'utilisation de la commande `\partial` en début de partition et d'une commande `\bar " | "` ou `\bar " | | "` à la fin de chaque ligne.

Modèle pour cantique

Le code ci-dessous illustre la manière d'agencer un cantique liturgique dans lequel chaque ligne débute et se termine par une mesure incomplète. Vous noterez par ailleurs l'affichage des paroles indépendamment de la musique.

```

Timeline = {
    \time 4/4
    \tempo 4=96
    \partial 2
    s2 | s1 | s2 \breathe s2 | s1 | s2 \bar " | | " \break
    s2 | s1 | s2 \breathe s2 | s1 | s2 \bar " | | "
}

SopranoMusic = \relative g' {
    g4 g | g g g g | g g g g | g g g g | g2
    g4 g | g g g g | g g g g | g g g g | g2
}

```

```

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic = \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
  \key g \major
}

\score { % Start score
  <<
    \new PianoStaff << % Start pianostaff
      \new Staff << % Start Staff = RH
        \global
        \clef "treble"
        \new Voice = "Soprano" << % Start Voice = "Soprano"
          \Timeline
          \voiceOne
          \SopranoMusic
        >> % End Voice = "Soprano"
        \new Voice = "Alto" << % Start Voice = "Alto"
          \Timeline
          \voiceTwo
          \AltoMusic
        >> % End Voice = "Alto"
      >> % End Staff = RH
    \new Staff << % Start Staff = LH
      \global
      \clef "bass"
      \new Voice = "Tenor" << % Start Voice = "Tenor"
        \Timeline
        \voiceOne
        \TenorMusic
      >> % End Voice = "Tenor"
      \new Voice = "Bass" << % Start Voice = "Bass"
        \Timeline
        \voiceTwo
        \BassMusic
      >> % End Voice = "Bass"
    >> % End Staff = LH
  >> % End pianostaff
  >>
}

```

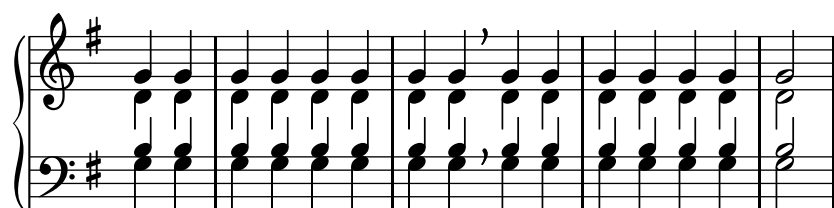
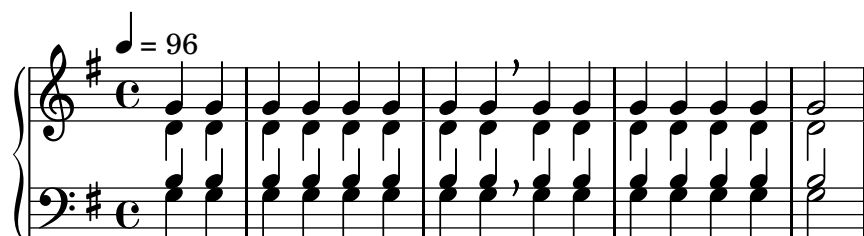
```

} % End score

\markup {
  \fill-line {
    ""
    {
      \column {
        \left-align {
          "This is line one of the first verse"
          "This is line two of the same"
          "And here's line three of the first verse"
          "And the last line of the same"
        }
      }
    }
  }
}

\paper { % Start paper block
  indent = 0 % don't indent first system
  line-width = 130 % shorten line length to suit music
} % End paper block

```



This is line one of the first verse
 This is line two of the same
 And here's line three of the first verse
 And the last line of the same

2.1.8 Musique vocale ancienne

LilyPond prend en charge la musique vocale ancienne. Elle est abordée en détails au chapitre [Section 2.9 \[Notations anciennes\]](#), page 407.

Voir aussi

Manuel de notation : [Section 2.9 \[Notations anciennes\]](#), page 407.

2.2 Instruments utilisant des portées multiples

Un peu retenu
très expressif

The musical score consists of four systems of staves. The first system is a grand staff (treble and bass clef) in 2/4 time, marked *ppp*. The second system is a grand staff with a *Rall.* marking, a *long* note, and a *ped.* marking. The third system is a grand staff with a *Rallentando* marking. The fourth system is a grand staff with a *Lent* marking, a *ppp* marking, and an *8va* marking. The score includes various musical notations such as notes, rests, and dynamic markings.

ppp

Rall. *long* *a Tempo* *pp* *ped.*

Rallentando

Lent *ppp* *8va*

Ce chapitre traite des différents aspects de la notation que l'on rencontre particulièrement avec les instruments qui ont recours à plusieurs portées, tels que ceux disposant de claviers, la harpe ou le vibraphone. Pour les besoins du discours, et pour simplifier, nous parlerons ici de « clavier » bien que le recours à des portées multiples concerne aussi des instruments qui en sont dépourvus.

2.2.1 Vue d'ensemble des claviers

Nous allons examiner ici les problèmes qui peuvent survenir en matière de notation pour la plupart des instrument à cordes multiples.

Généralités sur les instruments à clavier

La notation pour instrument à clavier est en règle générale présentée sous la forme d'un système pour piano, autrement dit deux portées normales ou plus réunies par une accolade. Cette notation sert également à la harpe ou à d'autres instruments à clés. L'organiste, quant à lui, lira une partition composée de deux portées au sein d'un système pianistique auquel vient s'adjoindre une portée normale pour le pédalier.

Les portées sont largement autonomes, mais il arrive que des voix passent de l'une à l'autre. Le contexte `PianoStaff` est précisément conçu pour gérer la notation spécifique au piano et autres instruments à clavier, notamment ces croisements.

Certaines particularités de la notation pour claviers sont abordées dans d'autres chapitres :

- Les claviers ont régulièrement recours à plusieurs voix dont le nombre peut varier. Voir à ce sujet [\[Résolution des collisions\]](#), page 158.
- On peut écrire la musique pour claviers de façon parallèle, comme l'explique [\[Saisie de musique en parallèle\]](#), page 168.
- Les nuances peuvent se gérer dans un contexte `Dynamics` qu'il suffira d'insérer entre les deux contextes `Staff` pour qu'elles apparaissent sur leur propre ligne centrée entre les deux portées. Voir à ce sujet [\[Nuances\]](#), page 112.
- Les indications de doigté sont abordées dans [\[Doigtés\]](#), page 201.
- Les indications en matière de pédalier d'orgue sont traitées comme des articulations. Voir à ce sujet [Section A.13 \[Liste des signes d'articulation\]](#), page 687.
- Pour générer des repères verticaux, voir [\[Quadrillage temporel\]](#), page 208.
- En plus des *Laissez vibrer*, les liaisons en matière de clavier peuvent intervenir sur des accords arpégés ou des trémolos. Reportez-vous au chapitre [\[Liaisons de prolongation\]](#), page 47.
- Le traitement des arpèges couvrant plusieurs voix ou portées est abordé au chapitre [\[Arpèges\]](#), page 129.
- Une description des indications de trémolo est disponible au chapitre [\[Répétitions en trémolo\]](#), page 147.
- Certaines retouches particulières au monde des claviers sont abordées au chapitre [Section "Exemple concret" dans Manuel d'initiation](#).
- Des notes fantômes permettent d'introduire des liaisons de tenue qui passent d'une voix à l'autre, comme le montre [Section "Autres utilisations des retouches" dans Manuel d'initiation](#).

Voir aussi

Manuel d'initiation : [Section "Exemple concret" dans Manuel d'initiation](#), [Section "Autres utilisations des retouches" dans Manuel d'initiation](#).

Manuel de notation : [\[Regroupement de portées\]](#), page 172, [\[Noms d'instrument\]](#), page 188, [\[Résolution des collisions\]](#), page 158, [\[Saisie de musique en parallèle\]](#), page 168, [\[Doigtés\]](#),

page 201, Section A.13 [Liste des signes d’articulation], page 687, [Quadrillage temporel], page 208, [Liaisons de prolongation], page 47, [Arpèges], page 129, [Répétitions en trémolo], page 147.

Morceaux choisis : Section “Claviers” dans *Morceaux choisis*.

Référence des propriétés internes : Section “PianoStaff” dans *Référence des propriétés internes*.

Changement de portée manuel

Il est possible de passer d’une portée à l’autre de façon manuelle, au moyen de la commande

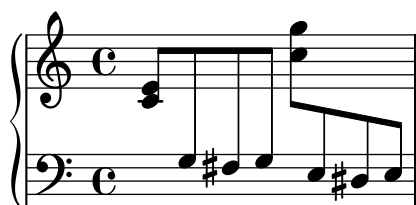
```
\change Staff = nomDeLaPortee
```

La valeur *nomDeLaPortee* est le nom de la portée sur laquelle va se déplacer la voix courante. Pour des raisons pratiques, on nomme la portée supérieure "haut" et la portée inférieure "bas", donc *nomDeLaPortee* désigne habituellement "haut", "bas", "MD" ou "MG".

Dans tous les cas, le contexte de portée ainsi utilisé doit exister au préalable. Le cas échéant, vous devrez « garder actives » certaines portées – voir Section 5.1.3 [Conservation d’un contexte], page 552 à ce propos.

Pour ligaturer automatiquement des notes entre deux portées, procédez ainsi :

```
\new PianoStaff <<
  \new Staff = "up" {
    <e' c'>8
    \change Staff = "down"
    g8 fis g
    \change Staff = "up"
    <g' c'>8
    \change Staff = "down"
    e8 dis e
    \change Staff = "up"
  }
  \new Staff = "down" {
    \clef bass
    % keep staff alive
    s1
  }
>>
```



Si les ligatures demandaient à être retouchées, commencez par modifier la direction des hampes. L’emplacement des ligatures sera alors calculé à partir du centre de la portée la plus proche – voir Section “Correction des collisions d’objets” dans *Manuel d’initiation* pour un exemple de retouche sur des ligatures.

Une voix qui change de portée peut entraîner des collisions :

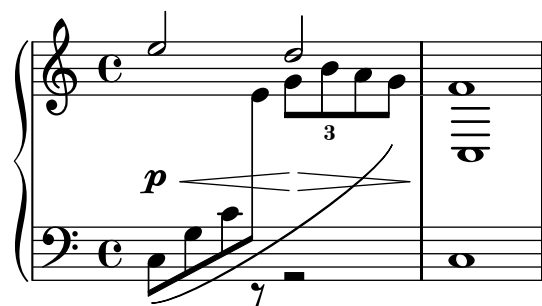
```
\new PianoStaff <<
  \new Staff = "up" {
    \voiceOne
```



```

% Make space for fingering in the cross-staff voice
\once\override DynamicLineSpanner #'staff-padding = #3.4
e''2\p\< d''\>
c1\!
}
\new Staff = "down" <<
{
  \clef bass
  s4. e,8\rest g,2\rest
  c1
} \ {
  c8\ ( g c'
  \change Staff = "up"
  e' g' b'-3 a' g'\ )
  f'1
}
>>
>>

```



Hampes et liaisons viennent en surimpression sur la ligne des nuances parce que la résolution automatique des collisions est inactivée pour ce qui relie les notes de différentes portées ainsi que pour les hampes ou extenseurs affectés à des notes incluses dans un changement de portée. Lorsque des collisions surviennent en pareil cas, il vous faudra les résoudre, en suivant les directives du chapitre [Section “Correction des collisions d’objets”](#) dans *Manuel d’initiation*.

Voir aussi

Manuel d’initiation : [Section “Correction des collisions d’objets”](#) dans *Manuel d’initiation*.

Manuel de notation : [\[Hampes\]](#), page 207, [\[Barres de ligature automatiques\]](#), page 75, [Section 5.1.3 \[Conservation d’un contexte\]](#), page 552.

Morceaux choisis : [Section “Claviers”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Beam”](#) dans *Référence des propriétés internes*, [Section “ContextChange”](#) dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Dans la mesure où l’évitement de collision des ligatures ne fonctionne pas lorsqu’une ligature automatique prend fin juste avant un changement de portée, utilisez alors une ligature manuelle.

Changement de portée automatique

Les voix peuvent passer automatiquement d’une portée à l’autre, au moyen de la syntaxe suivante :

```
\autochange ...musique...
```

Deux portées seront alors créées au sein du contexte `PianoStaff`, nommées respectivement "up" et "down". La portée du bas sera par défaut en clef de fa. La commande `\autochange` bascule les notes d'une portée à l'autre en fonction de leur hauteur (le do du milieu servant de pivot), et place les silences en fonction des notes qui les suivront. Ainsi :

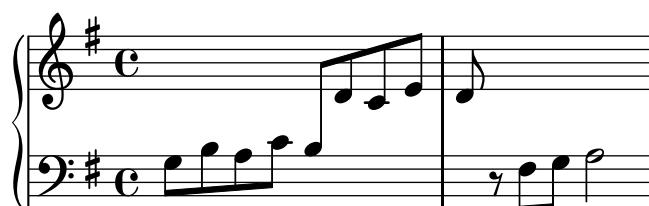
```
\new PianoStaff {
  \autochange {
    g4 a b c'
    d'4 r a g
  }
}
```



Une section en mode `\relative` se situant en dehors de la commande `\autochange` n'aura pas d'effet sur les hauteurs de l'expression *musique*. Il est donc préférable d'introduire la directive `\relative` après `\autochange`.

Lorsque des contrôles particuliers doivent s'appliquer aux portées, mieux vaut les nommer explicitement – attention : sous peine d'effet indésirable quant au résultat, la portée supérieure doit s'appeler "up" et l'inférieure "down" ; *ceci est à notre connaissance le seul cas où ces noms de variable sont figés*. Cette procédure sert, entre autres, à indiquer l'armure sur la portée inférieure :

```
\new PianoStaff <<
  \new Staff = "up" {
    \new Voice = "melodieUn" {
      \key g \major
      \autochange \relative c' {
        g8 b a c b d c e
        d8 r fis, g a2
      }
    }
  }
  \new Staff = "down" {
    \key g \major
    \clef bass
  }
}>>
```



Voir aussi

Manuel de notation : [Changement de portée manuel], page 302.

Morceaux choisis : Section “Claviers” dans *Morceaux choisis*.

Référence des propriétés internes : Section “AutoChangeMusic” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les changements de portée automatiques n'interviennent pas toujours à l'endroit le plus opportun. Pour un résultat de meilleure qualité, il vaut mieux indiquer soi-même ces changements.

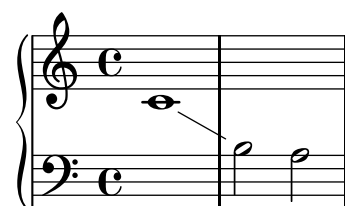
Un accord ne peut se répartir sur plusieurs portées ; sa portée d'affectation sera déterminée par la première hauteur mentionnée dans la construction de cet accord.

`\autochange` ne peut intervenir à l'intérieur d'une commande `\times`.

Lignes de changement de portée

Lorsqu'une voix change de portée, il est possible d'imprimer automatiquement une ligne reliant les notes, en faisant appel à la commande `\showStaffSwitch` :

```
\new PianoStaff <<
  \new Staff = "one" {
    \showStaffSwitch
    c1
    \change Staff = "two"
    b2 a
  }
  \new Staff = "two" {
    \clef bass
    s1*2
  }
>>
```



Commandes prédéfinies

`\showStaffSwitch`, `\hideStaffSwitch`.

Voir aussi

Morceaux choisis : Section “Claviers” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Note_head_line_engraver” dans *Référence des propriétés internes*, Section “VoiceFollower” dans *Référence des propriétés internes*.

Hampes et changements de portée

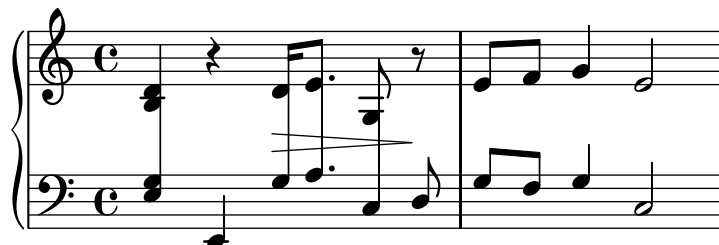
Pour écrire des accords qui enjambent deux portées, on allonge, à l'aide du `Span_stem_engraver`, la hampe de l'accord de l'une des deux portées de façon à ce qu'elle rejoigne celle de l'autre portée. La gestion des ligatures automatiques requiert alors une attention particulière ; en effet, une seule portée devrait alors comporter des ligatures.

```

\layout {
  \context {
    \PianoStaff
    \consists #Span_stem_engraver
  }
}

{
  \new PianoStaff <<
    \new Staff {
      <b d'>4 r d'16\> e'8. g8 r\!
      e'8 f' g'4 e'2
    }
    \new Staff {
      \clef bass
      \voiceOne
      \autoBeamOff
      \crossStaff { <e g>4 e, g16 a8. c8} d
      \autoBeamOn
      g8 f g4 c2
    }
  >>
}

```



Il n'est pas possible à l'heure actuelle, en raison de la manière dont il a été implémenté, de spécifier ce graveur en le mettant entre guillemets ; il faut le mentionner en faisant précéder son nom d'un symbole *hash* (un #).

Morceaux choisis

Indication d'un accord à cheval sur deux portées par un crochet d'arpège

Un crochet d'arpège peut indiquer que des notes réparties sur deux portées différentes doivent être jouées par la même main. Le contexte `PianoStaff` doit accepter des arpèges « distribués », et les indications d'arpège du contexte `PianoStaff` adopter une allure de crochet.

(Debussy, Les collines d'Anacapri, mesure 65)

```

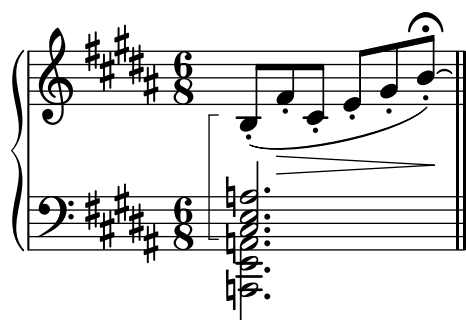
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \override PianoStaff.Arpeggio #'stencil = #ly:arpeggio::brew-chord-bracket
  \new Staff {
    \relative c' {
      \key b \major
      \time 6/8
      b8-.(\arpeggio fis'-.\> cis-. e-. gis-. b-.)\!\fermata^\laissezVibrer
      \bar "||"
    }
  }
  >>

```

```

    }
  }
  \new Staff {
    \relative c' {
      \clef bass
      \key b \major
      <<
      {
        <a e cis>2.\arpeggio
      }
      \\
      {
        <a, e a,>2.
      }
      >>
    }
  }
  >>

```



Voir aussi

Morceaux choisis : [Section “Claviers”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “Stem”](#) dans *Référence des propriétés internes*.

2.2.2 Piano

Ce chapitre traite des aspects de la notation directement liés au piano.

Pédales de piano

Le piano possède deux pédales, parfois trois, permettant de modifier l'émission du son : une pédale de *tenue* (*sustain*), une pédale de *sourdine* (*una corda* ou *U.C.*) et une pédale *tonale* (*sostenuto* ou *sos.*). La pédale *sustain* se rencontre aussi sur les vibraphones et celestas.

```

c4\sustainOn d e g
<c, f a>1\sustainOff
c4\sostenutoOn e g c,
<bes d f>1\sostenutoOff
c4\unaCorda d e g
<d fis a>1\treCorde

```



Trois styles sont à votre disposition pour imprimer les indications de pédale : sous forme de texte, de crochet, ou une combinaison des deux. `text` est le style de notation par défaut pour les pédales de tenue ou de sourdine — le traditionnel « *Ped. ». La pédale tonale, en revanche, utilise `mixed` par défaut.

```
c4\sustainOn g c2\sustainOff
\set Staff.pedalSustainStyle = #'mixed
c4\sustainOn g c d
d\sustainOff\sustainOn g, c2\sustainOff
\set Staff.pedalSustainStyle = #'bracket
c4\sustainOn g c d
d\sustainOff\sustainOn g, c2
\bar "|."
```



Le placement des commandes de pédale correspond au mouvement de la pédale de tenue pendant le jeu. Garder une pédale enclenchée jusqu'à la barre finale s'obtient tout simplement en omettant la commande de relâcher.

Les indication de pédale peuvent s'inscrire dans un contexte `Dynamics`, ce qui aura pour effet de leur attribuer une ligne en propre.

Voir aussi

Manuel de notation : [Liaisons de prolongation], page 47.

Morceaux choisis : Section “Claviers” dans *Morceaux choisis*.

Référence des propriétés internes : Section “SustainPedal” dans *Référence des propriétés internes*, Section “SustainPedalLineSpanner” dans *Référence des propriétés internes*, Section “SustainEvent” dans *Référence des propriétés internes*, Section “SostenutoPedal” dans *Référence des propriétés internes*, Section “SostenutoPedalLineSpanner” dans *Référence des propriétés internes*, Section “SostenutoEvent” dans *Référence des propriétés internes*, Section “UnaCordaPedal” dans *Référence des propriétés internes*, Section “UnaCordaPedalLineSpanner” dans *Référence des propriétés internes*, Section “UnaCordaEvent” dans *Référence des propriétés internes*, Section “PianoPedalBracket” dans *Référence des propriétés internes*, Section “PianoPedalEngraver” dans *Référence des propriétés internes*.

2.2.3 Accordéon

Symboles de jeux

De nombreux accordéons possèdent plusieurs jeux d'anches capables de jouer à l'unisson ou bien à l'octave inférieure ou supérieure par rapport aux notes écrites. Chaque facteur d'accordéon donne des noms différents aux *tirettes* (*shifts*) qui permettent de sélectionner les différents jeux d'anches tels que *hautbois*, *musette* ou *bandonéon*, de telle sorte qu'un système de symbole a fini par voir le jour afin de faciliter les instructions pour exécuter un morceau.

Morceaux choisis

Symboles de chant pour accordéon

Les symboles spécifiques au clavier main droite d'un accordéon s'indiquent à l'aide de `\markup`. Le positionnement de tels symboles s'ajuste grâce à des arguments `\raise`.

```

discant = \markup {
  \musicglyph #"accordion.discant"
}
dot = \markup {
  \musicglyph #"accordion.dot"
}

\layout { ragged-right = ##t }

% 16 voets register
accBasson = ^\markup {
  \combine
  \discant
  \raise #0.5 \dot
}

% een korig 8 en 16 voets register
accBandon = ^\markup {
  \combine
  \discant
  \combine
  \raise #0.5 \dot
  \raise #1.5 \dot
}

accVCello = ^\markup {
  \combine
  \discant
  \combine
  \raise #0.5 \dot
  \combine
  \raise #1.5 \dot
  \translate #'(1 . 0) \raise #1.5 \dot
}

% 4-8-16 voets register
accHarmon = ^\markup {
  \combine
  \discant
  \combine
  \raise #0.5 \dot
  \combine
  \raise #1.5 \dot
  \raise #2.5 \dot
}

accTrombon = ^\markup {
  \combine
  \discant
  \combine
  \raise #0.5 \dot
  \combine

```

```

        \raise #1.5 \dot
        \combine
        \translate #'(1 . 0) \raise #1.5 \dot
        \translate #'(-1 . 0) \raise #1.5 \dot
    }

% eenkorig 4 en 16 voets register
accOrgan = ^\markup {
    \combine
    \discant
    \combine
    \raise #0.5 \dot
    \raise #2.5 \dot
}

accMaster = ^\markup {
    \combine
    \discant
    \combine
    \raise #0.5 \dot
    \combine
    \raise #1.5 \dot
    \combine
    \translate #'(1 . 0) \raise #1.5 \dot
    \combine
    \translate #'(-1 . 0) \raise #1.5 \dot
    \raise #2.5 \dot
}

accAccord = ^\markup {
    \combine
    \discant
    \combine
    \raise #1.5 \dot
    \combine
    \translate #'(1 . 0) \raise #1.5 \dot
    \combine
    \translate #'(-1 . 0) \raise #1.5 \dot
    \raise #2.5 \dot
}

accMusette = ^\markup {
    \combine
    \discant
    \combine
    \raise #1.5 \dot
    \combine
    \translate #'(1 . 0) \raise #1.5 \dot
    \translate #'(-1 . 0) \raise #1.5 \dot
}

accCeleste = ^\markup {

```



```

\combine
\discant
\combine
\raise #1.5 \dot
\translate #'(-1 . 0) \raise #1.5 \dot
}

accOboe = ^\markup {
\combine
\discant
\combine
\raise #1.5 \dot
\raise #2.5 \dot
}

accClarin = ^\markup {
\combine
\discant
\raise #1.5 \dot
}

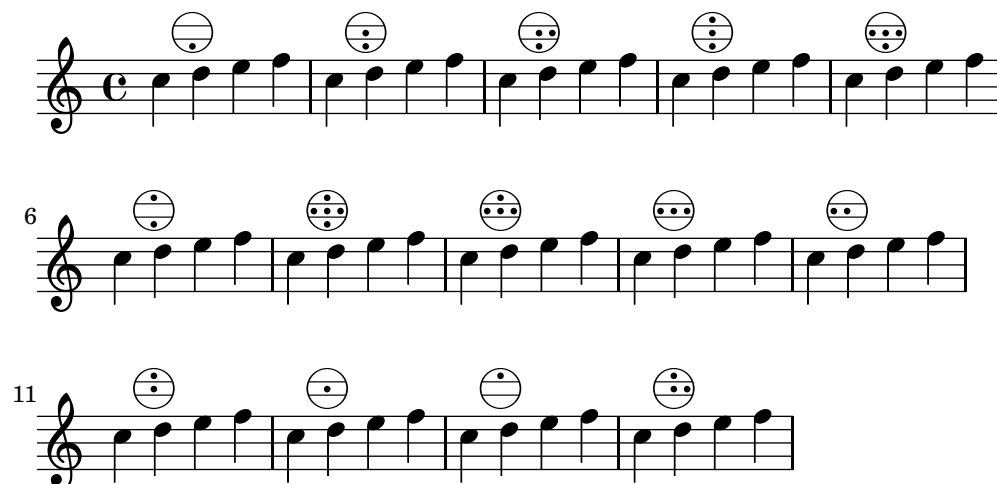
accPiccolo = ^\markup {
\combine
\discant
\raise #2.5 \dot
}

accViolin = ^\markup {
\combine
\discant
\combine
\raise #1.5 \dot
\combine
\translate #'(1 . 0) \raise #1.5 \dot
\raise #2.5 \dot
}

\relative c'' {
c4 d\accBasson e f
c4 d\accBandon e f
c4 d\accVCello e f
c4 d\accHarmon e f
c4 d\accTrombon e f
\break
c4 d\accOrgan e f
c4 d\accMaster e f
c4 d\accAccord e f
c4 d\accMusette e f
c4 d\accCeleste e f
\break
c4 d\accOboe e f
c4 d\accClarin e f

```

```
c4 d\accPiccolo e f
c4 d\accViolin e f
}
```



Voir aussi

Morceaux choisis : [Section “Keyboards, Claviers”](#) dans *Morceaux choisis*.

2.2.4 Harpe

Cette partie s'intéresse aux particularités en matière de notation pour la harpe.

Généralités sur la harpe

Certaines caractéristiques de la musique pour harpes sont abordées dans d'autres chapitres, tels que :

- Les glissandos, l'une des techniques spécifique à la harpe, [\[Glissando\]](#), page 125.
- Le *bisbigliando*, qui s'écrit comme un trémolo, [\[Répétitions en trémolo\]](#), page 147.
- Les harmoniques naturelles sont présentées dans [\[Harmoniques\]](#), page 315.
- L'interprétation des arpeggios est abordée dans [\[Arpèges\]](#), page 129.

Voir aussi

Manuel de notation : [\[Répétitions en trémolo\]](#), page 147, [\[Glissando\]](#), page 125, [\[Arpèges\]](#), page 129, [\[Harmoniques\]](#), page 315.

Pédales de harpe

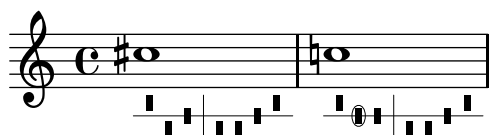
Les harpes comportent sept cordes par octave qui peuvent sonner naturel, dièse ou bémol. Si chacune des cordes de la harpe celtique (*lever harp*) est accordée individuellement, celles d'une harpe à pédalier ayant la même note de base sont contrôlées par une seule pédale. De gauche à droite, elles correspondent aux notes ré, do, si, et mi, fa, sol, la pour la grande harpe. Les trois premières pédales sont réservées au pied gauche, les quatre dernières au pied droit. Leur position peut être indiquée par une marque textuelle :

```
\textLengthOn
cis1_\markup \concat \vcenter {
  [D \flat C \sharp B|E \sharp F \sharp G A \flat] }
c!1_\markup \concat \vcenter {
  [ C \natural ]}
```



ou bien sous forme de diagramme :

```
\textLengthOn
cis1_\markup { \harp-pedal #"^v-|vv-^" }
c!1_\markup { \harp-pedal #"^o--|vv-^" }
```



Bémol si la pédale est relâchée (ou en haut), bécarré si elle est bloquée sur le cran du milieu, et dièse si elle est tout à fait enfoncée. La commande `\harp-pedal` prend en argument une chaîne de caractères, où `^` indique que la pédale est relâchée ou en haut (bémol), `-` qu'elle est bloquée sur le cran du milieu (bécarré), `v` qu'elle est tout à fait enfoncée (dièse) et `|` représente le séparateur (entre gauche et droite de l'instrumentiste). Faire précéder un symbole par un `o` permet de l'inscrire dans un cercle.

Voir aussi

Manuel de notation : [Commentaires textuels], page 213, Section A.10.5 [Instrument Specific Markup], page 675.

2.3 Cordes non frettées

The image displays three staves of musical notation for non-fretted strings, illustrating various performance techniques and dynamics.

Staff 1: Marked *lentement* and *fatigué*. It shows a sequence of notes with dynamic markings *mf*, *ff*, and *pp*. Above the staff, there are articulation marks including *s. vib.* (sustained vibration) and *p. vib.* (pizzicato vibration). The notes are connected by slurs, and there are various bowing or playing marks like *IV* and *0*.

Staff 2: Marked *accél...* (accelerando). It shows a sequence of notes with dynamic markings *mf* and *ff*. Above the staff, there are articulation marks including *s.p.* (sustained pizzicato) and *n.* (normal). The notes are connected by slurs, and there are various bowing or playing marks like *IV* and *0*.

Staff 3: Marked *ritar...* (ritardando). It shows a sequence of notes with dynamic markings *ppp* (pianissimo). Above the staff, there are articulation marks including *s.p.*, *n.*, and *m. vib.* (moderate vibration). The notes are connected by slurs, and there are various bowing or playing marks like *IV* and *0*.

Cette section dispense des informations supplémentaires et utiles à l'écriture pour les cordes non frettées, et tout spécialement pour les cordes d'orchestre.

2.3.1 Vue d'ensemble de la notation pour cordes non frettées

Il existe peu d'éléments spécifiques à l'écriture pour instruments à cordes non frettées. La musique est notée sur une portée et ne nécessite généralement qu'une seule voix. Le recours à deux voix distinctes peut cependant permettre de traiter efficacement des doubles notes ou des divisions.

Références en matière de cordes non frettées

La majorité des éléments de notation nécessaires en matière de cordes d'orchestre et autres instruments à archet est abordée dans d'autres chapitres de la documentation :

- Les indications textuelles telles que « pizz. » ou « arco » sont ajoutées comme du texte simple – voir à ce sujet [Commentaires textuels], page 213.
- Les indications de doigtés, incluant les indications du pouce, sont décrites dans [Doigtés], page 201.
- Les doubles notes sont généralement indiquées en écrivant un accord, voir [Notes en accords], page 149. Les précisions pour jouer des accords peuvent être ajoutées, comme l'indique [Arpèges], page 129.
- Un modèle pour quatuor à cordes est disponible à la rubrique Section “Quatuor à cordes” dans *Manuel d'initiation*. D'autres informations se trouvent dans les exemples de code.

Voir aussi

Manuel d'initiation : Section “Quatuor à cordes” dans *Manuel d'initiation*.

Manuel de notation : [Commentaires textuels], page 213, [Doigtés], page 201, [Notes en accords], page 149, [Arpèges], page 129.

Morceaux choisis : Section “Cordes non frettées” dans *Morceaux choisis*.

Indications d'archet

Les indications d'archet se créent comme des articulations, elles sont décrites dans [Articulations et ornements], page 109.

Les indications d'archet, poussé (`\upbow`) et tiré (`\downbow`), peuvent se combiner à des liaisons comme ici :

```
c4(\downbow d) e(\upbow f)
```

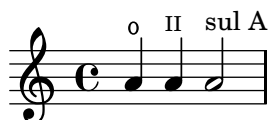


ou dans l'exemple suivant qui montre trois manières d'indiquer la corde de La, à vide, sur un violon :

```
a4 \open
```

```
a^\markup { \teeny "II" }
```

```
a2^\markup { \small "sul A" }
```



Commandes prédéfinies

`\downbow`, `\upbow`, `\open`.

Voir aussi

Manuel de notation : [\[Articulations et ornements\]](#), page 109, [\[Liaisons d'articulation\]](#), page 119.

Harmoniques

Harmoniques naturels

Les harmoniques naturels sont indiqués de différentes manières : une tête de note en forme de losange signifie généralement que vous devez effleurer la corde à l'endroit où vous l'auriez pincée si la note avait été normale.

```
d4 e4.
\harmonicsOn
d8 e e
d4 e4.
\harmonicsOff
d8 e e
```



Une autre façon de procéder consiste à faire surmonter la note normale d'un petit cercle. Ceci indique que la note écrite doit être jouée en harmonique :

```
d2^\flageolet d_\flageolet
```



Un plus petit cercle peut être créé, comme indiqué dans les exemples de code contenus dans [\[Références en matière de cordes non frettes\]](#), page 314.

Harmoniques artificiels

Les harmoniques artificiels sont indiqués par une double tête de note : l'une normale, indique la note à pincer, et l'autre, en forme de losange, indique l'endroit où la corde doit être effleurée.

La propriété `harmonicDots`, lorsqu'elle est activée, permet d'ajouter un point aux notes pointées affublées d'un `\harmonic`.

```
<e a\harmonic>2. <c g'\harmonic>4
\set harmonicDots = ##t
<e a\harmonic>2. <c g'\harmonic>4
```



Note : La commande `\harmonic` doit être placée dans une construction de type accord même s'il n'y a qu'une seule note. On utiliserait plutôt la commande `\harmonicsOn` en pareil cas.

Voir aussi

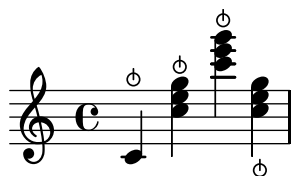
Glossaire musicologique : [Section “harmonics”](#) dans *Glossaire*.

Manuel de notation : [Têtes de note spécifiques], page 33, [Références en matière de cordes non frettées], page 314.

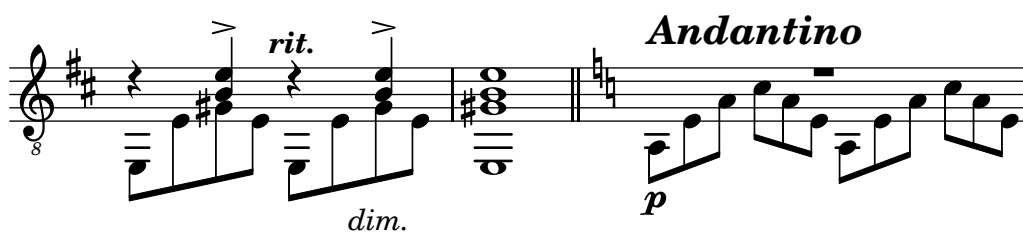
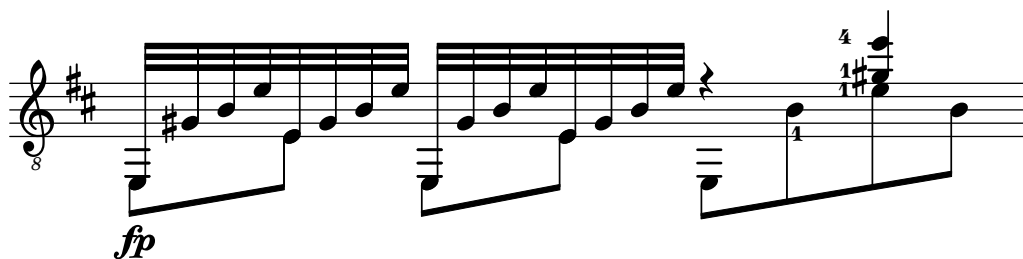
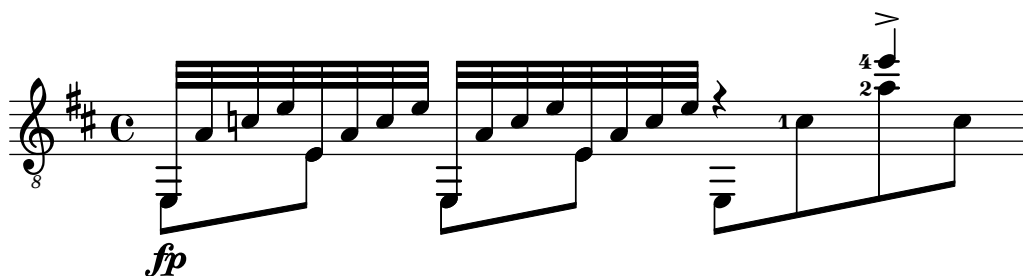
Snap (Bartók) pizzicato

Un *snap pizzicato*, aussi appelé « Bartok pizz » est un type de pizzicato pour lequel la corde est tirée vers le haut (plutôt que sur le côté) de telle sorte qu'elle vienne frapper le manche.

```
c4\snappizzicato
<c' e g>4\snappizzicato
<c' e g>4^\snappizzicato
<c, e g>4_\snappizzicato
```



2.4 Instruments à cordes frettées





Cette section traite de différents aspects de la notation propre aux instruments à cordes frettées.

2.4.1 Vue d'ensemble des cordes frettées

Nous allons aborder, dans les paragraphes qui suivent, les particularités communes aux différents instruments à cordes frettées.

Références en matière de cordes frettées

La musique pour instruments à cordes frettées est généralement notée sur une seule portée, en notation traditionnelle ou en tablature, les deux étant parfois combinées. Il est aussi courant en musique populaire d'utiliser des diagrammes d'accord au-dessus de la portée traditionnelle. La guitare et le banjo sont des instruments transpositeurs, sonnant une octave au-dessous de ce qui est écrit. Les partitions pour ces instruments utilisent donc la clé de sol ottava bassa "treble_8" – ou une instruction `\transposition c` pour un rendu MIDI correct. Vous pourrez trouver ailleurs dans la documentation d'autres éléments aussi utilisés par les instruments à cordes frettées :

- Les doigtés s'obtiennent comme indiqué au chapitre [Doigtés], page 201.
- En plus des *Laissez vibrer*, les liaisons peuvent intervenir sur des accords arpégés ou des trémolos. Reportez-vous au chapitre [Liaisons de prolongation], page 47.
- Des indications quant au support polyphonique se trouvent au chapitre [Résolution des collisions], page 158.
- La notation des sons harmoniques se trouve à la section [Harmoniques], page 315.

Voir aussi

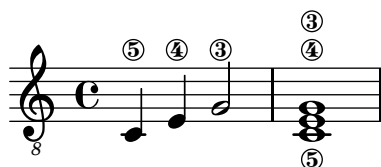
Manuel de notation : [Doigtés], page 201, [Liaisons de prolongation], page 47, [Résolution des collisions], page 158, [Noms d'instrument], page 188, [Saisie de musique en parallèle], page 168, [Arpèges], page 129, Section A.13 [Liste des signes d'articulation], page 687, [Clefs], page 16.

Indications du numéro de corde

Une corde sur laquelle une note doit être jouée peut être indiquée en attachant `\numéro` à cette note prise dans une construction de type accord `<>`.

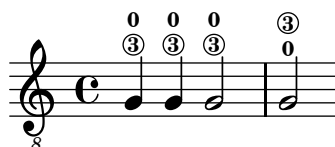
Note : Les numéros de corde **doivent** être définis dans une construction de type accord même s'il n'y a qu'une seule note.

```
\clef "treble_8"
c4\5 e\4 g2\3
<c,\5 e\4 g\3>1
```



Quand les indications de doigté et de numéro de corde sont attachées à une même note, leur positionnement se fera en fonction de l'ordre dans lequel elles apparaissent dans le code et **uniquement** si elles interviennent au sein d'une construction d'accord : le positionnement des doigtés est géré différemment selon qu'ils s'appliquent à l'accord entier ou à des notes isolées indépendantes d'un accord.

```
\clef "treble_8"
g4\3-0
g-0\3
<g\3-0>2
<g-0\3>
```

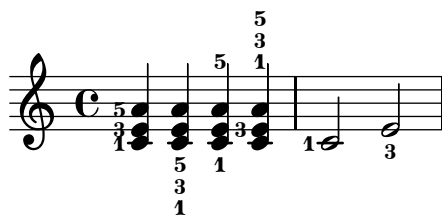


Morceaux choisis

Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord < >, même s'il ne s'agit que d'une seule note.

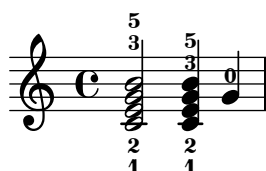
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
}
```

Impression des doigtés à l'intérieur de la portée

L'empilement des indications de doigté se fait par défaut à l'extérieur de la portée. Néanmoins, il est possible d'annuler ce comportement.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering #'staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 <g'-0>
}
```



Voir aussi

Manuel de notation : [\[Doigtés\]](#), page 201.

Morceaux choisis : [Section “Cordes frettées”](#) dans *Morceaux choisis*.

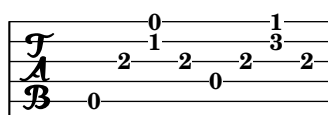
Référence des propriétés internes : [Section “StringNumber”](#) dans *Référence des propriétés internes*, [Section “Fingering”](#) dans *Référence des propriétés internes*.

Tablatures par défaut

La notation en tablature est utilisée pour certains instruments à cordes pincées. Les hauteurs n'y sont pas indiquées par des têtes de note, mais par des chiffres ou autres symboles qui indiquent sur quelle corde et à quelle case chaque note doit être jouée. Des notes devant être jouées simultanément seront alors alignées verticalement.

Par défaut, la première corde est la plus aiguë et correspond à la ligne supérieure du `TabStaff`. Les cordes suivent par défaut l'accordage traditionnel d'une guitare (à six cordes). Les notes sont imprimées sous forme de tablature, dans les contextes `TabStaff` et `TabVoice` qui comportent une clef spécifique ajoutée automatiquement.

```
\new TabStaff \relative c' {
  a,8 a' <c e> a
  d,8 a' <d f> a
}
```



Par défaut, les tablatures ne comportent aucune marque de durée ni de symbole musical tel que des nuances.

```
symbols = {
  \time 3/4
  c4-.^"Allegro" d( e)
  f4-. \f g a^\fermata
}
```

```

\mark \default
c8_.\<\( c16 c ~ c2\!
c'2.\prall\
}

\score {
  <<
    \new Staff { \clef "G_8" \symbols }
    \new TabStaff { \symbols }
  >>
}

```

Pour obtenir les symboles de notation habituelle dans une tablature, il faut appliquer la commande `\tabFullNotation` au contexte `TabStaff`. Vous noterez par ailleurs que les blanches sont affublées d'une double hampe afin de ne pas les confondre avec des noires.

```

symbols = {
  \time 3/4
  c4-.\^"Allegro" d( e)
  f4-.\f g a^\fermata
  \mark \default
  c8_.\<\( c16 c ~ c2\!
  c'2.\prall\
}

```

```

\score {
  \new TabStaff {
    \tabFullNotation
    \symbols
  }
}

```

Quand aucune corde n'est précisée, LilyPond choisit automatiquement la corde pour laquelle la position est la moins élevée, avec une préférence pour une corde à vide. Vous pouvez préférer qu'une note donnée soit jouée sur une corde particulière, auquel cas l'affectation directe du numéro de corde suffit. L'absence d'indication des numéros de corde en notation traditionnelle

se gère au niveau des stencils. Il est cependant plus facile de jouer sur la propriété `minimumFret`, dont la valeur par défaut est fixée à 0, ce qui correspond aux cordes à vide.

Cependant, et en dépit d'une affectation de `minimumFret`, une corde à vide aura toujours préséance. Ce comportement se modifie par l'activation de `restrainOpenStrings`.

```
\layout { \override Voice.StringNumber #'stencil = ##f }
\new StaffGroup <<
  \new Staff \relative c {
    \clef "treble_8"
    \time 2/4
    c16 d e f g4
    c,16\5 d\5 e\4 f\4 g4\4
    c,16 d e f g4
  }
  \new TabStaff \relative c {
    c16 d e f g4
    c,16\5 d\5 e\4 f\4 g4\4
    \set TabStaff.minimumFret = #5
    \set TabStaff.restrainOpenStrings = ##t
    c,16 d e f g4
  }
>>
```

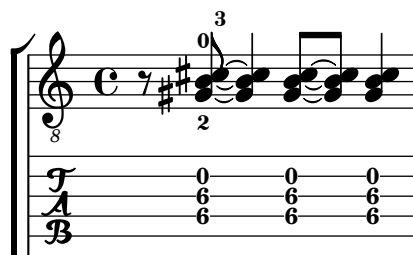
La répétition d'une construction en accord s'indique par un `q` – voir [\[Répétition d'accords\]](#), [page 151](#). Cette fonctionnalité, bien qu'opérationnelle en mode tablature, supprime entre autres les numéros de corde et doigtés. Il vous faudra donc préalablement recourir explicitement à l'instruction

```
\chordRepeats #'(string-number-event fingering-event)
```

si vous utilisez la répétition d'accords dans vos tablatures. Cette instruction est abrégée en `\tabChordRepeats`.

```
guitar = \relative c' {
  r8 <gis-2 cis-3 b-0>~ q4 q8~ q q4
}
```

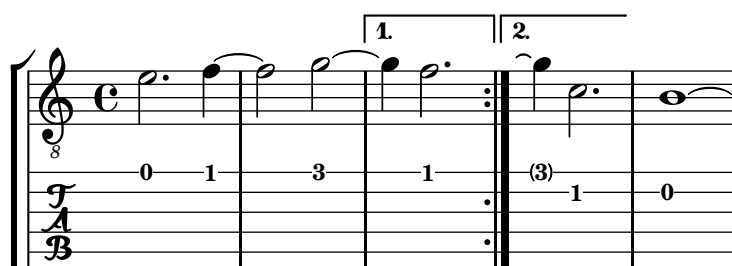
```
\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \guitar
  }
  \new TabStaff {
    \tabChordRepeats \guitar
  }
>>
```

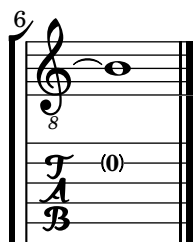


Lorsqu'une liaison de prolongation intervient à l'occasion d'un saut de ligne, la note est répétée, entre parenthèses. Il en va de même pour la seconde alternative d'une répétition.

```
ties = \relative c' {
  \repeat volta 2 {
    e2. f4~
    f2 g2~
  }
  \alternative {
    { g4 f2. }
    { g4\repeatTie c,2. }
  }
  b1~
  \break
  b1
  \bar "|"
}
```

```
\score {
  <<
    \new StaffGroup <<
      \context Staff {
        \clef "treble_8"
        \ties
      }
      \context TabStaff {
        \ties
      }
    >>
  >>
  \layout {
    indent = #0
    ragged-right = ##t
  }
}
```

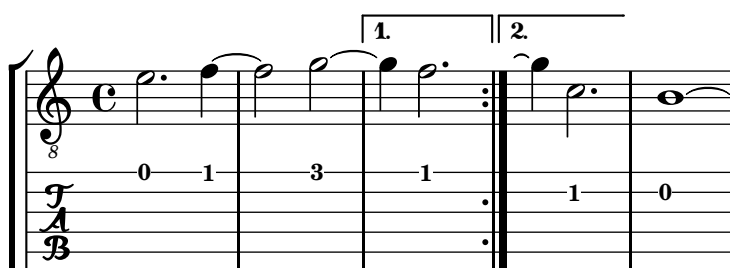


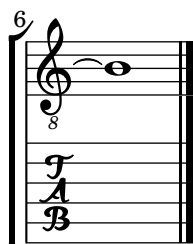


La commande `\hideSplitTiedTabNotes` permet d'éviter d'imprimer ces cases entre parenthèses.

```
ties = \relative c' {
  \repeat volta 2 {
    e2. f4~
    f2 g2~ }
  \alternative {
    { g4 f2. }
    { g4\repeatTie c,2. }
  }
  b1~
  \break
  b1
  \bar "|."
}

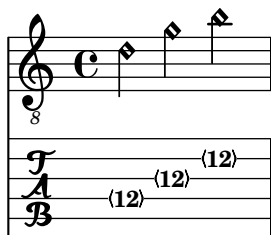
\score {
  <<
    \new StaffGroup <<
      \context Staff {
        \clef "treble_8"
        \ties
      }
      \context TabStaff {
        \hideSplitTiedTabNotes
        \ties
      }
    >>
  >>
  \layout {
    indent = #0
    ragged-right = ##t
  }
}
```





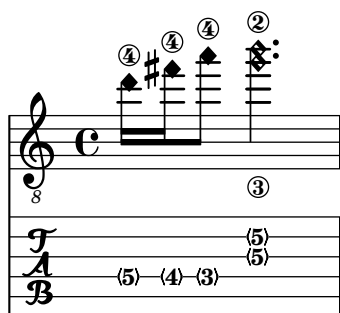
Les indications de sons harmoniques ainsi que les glissandos peuvent être ajoutés aux tablatures.

```
\layout { \override Voice.StringNumber #'stencil = ##f }
firstHarmonic = {
  d'4\4\harmonic
  g'4\3\harmonic
  b'2\2\harmonic
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \firstHarmonic
    }
    \new TabStaff { \firstHarmonic }
  >>
}
```



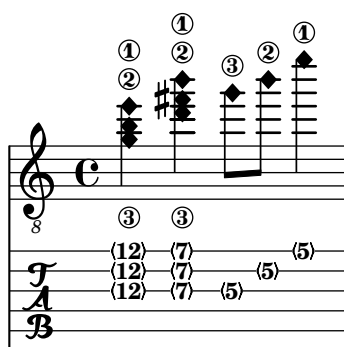
Vous noterez que la commande `\harmonic` s'attache toujours à une note unique (parfois contenue dans un accord) et non à l'ensemble d'un accord. Ceci ne fonctionne donc que pour une harmonique au douzième fret d'une corde à vide. Toute autre harmonique devrait être définie directement par LilyPond. Vous pouvez l'y aider en indiquant la case où le doigt viendrait se placer sur le manche.

```
fretHarmonics = {
  \harmonicByFret #5 d16\4
  \harmonicByFret #4 d16\4
  \harmonicByFret #3 d8\4
  \harmonicByFret #5 <g\3 b\2>2.
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \fretHarmonics
    }
    \new TabStaff { \fretHarmonics }
  >>
}
```



Une harmonique peut accessoirement se calculer à partir de la longueur de corde par rapport au doigté de cette harmonique.

```
ratioHarmonics = {
  \harmonicByRatio #1/2 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/3 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/4 { g8\3 b8\2 e'4\1 }
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \ratioHarmonics
    }
    \new TabStaff { \ratioHarmonics }
  >>
}
```

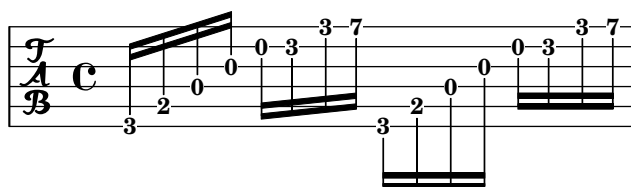


Morceaux choisis

Hampes et ligatures en mode tablature

La direction des hampes se gère dans les tablatures tout comme en notation traditionnelle. Les ligatures peuvent être mises à l'horizontale comme le montre cet exemple.

```
\new TabStaff {
  \relative c {
    \tabFullNotation
    g16 b d g b d g b
    \stemDown
    \override Beam #'concaveness = #10000
    g,,16 b d g b d g b
  }
}
```



Polyphonie en mode tablature

Une section polyphonique s'obtient dans un `TabStaff` de la même manière que dans une portée normale.

```
upper = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}
```

```
lower = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}
```

```
\score {
  <<
    \new StaffGroup = "tab with traditional" <<
      \new Staff = "guitar traditional" <<
        \clef "treble_8"
        \context Voice = "upper" \upper
        \context Voice = "lower" \lower
      >>
    \new TabStaff = "guitar tab" <<
      \context TabVoice = "upper" \upper
      \context TabVoice = "lower" \lower
    >>
  >>
}>>
```

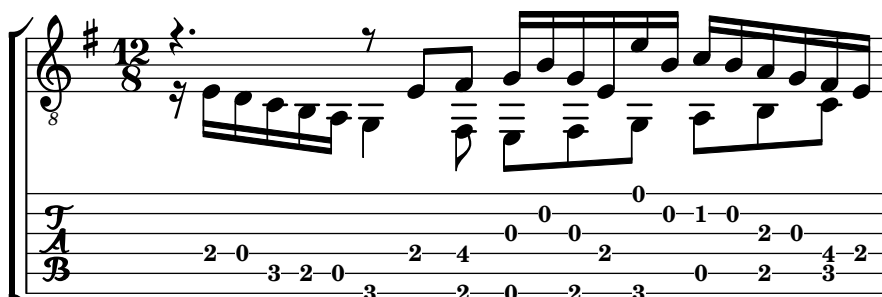


Table des harmoniques sur corde à vide

Table des harmoniques sur corde à vide (harmoniques naturelles) :

```
openStringHarmonics = {
  %first harmonic
  \harmonicByFret #12 e,2\6_\markup{"1st harm."}
  \harmonicByRatio #1/2 e,\6
```



```

%second harmonic
\harmonicByFret #7 e,\6_\markup{"2nd harm. - - - -"}
\harmonicByRatio #1/3 e,\6
\harmonicByFret #19 e,\6
\harmonicByRatio #2/3 e,\6
%\harmonicByFret #19 < e,\6 a,\5 d\4 >
%\harmonicByRatio #2/3 < e,\6 a,\5 d\4 >
%third harmonic
\harmonicByFret #5 e,\6_\markup{"3rd harm. - - - -"}
\harmonicByRatio #1/4 e,\6
\harmonicByFret #24 e,\6
\harmonicByRatio #3/4 e,\6
\break
%fourth harmonic
\harmonicByFret #4 e,\6_\markup{"4th harm. - - - - - - - - - -"}
\harmonicByRatio #1/5 e,\6
\harmonicByFret #9 e,\6
\harmonicByRatio #2/5 e,\6
\harmonicByFret #16 e,\6
\harmonicByRatio #3/5 e,\6
%fifth harmonic
\harmonicByFret #3 e,\6_\markup{"5th harm."}
\harmonicByRatio #1/6 e,\6
\break
%sixth harmonic
\harmonicByFret #2.7 e,\6_\markup{"6th harm."}
\harmonicByRatio #1/7 e,\6
%seventh harmonic
\harmonicByFret #2.3 e,\6_\markup{"7th harm."}
\harmonicByRatio #1/8 e,\6
%eighth harmonic
\harmonicByFret #2 e,\6_\markup{"8th harm."}
\harmonicByRatio #1/9 e,\6
}

\score {
  <<
    \new Staff {
      \new Voice {
        \clef "treble_8"
        \openStringHarmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \openStringHarmonics
      }
    }
  >>
}

```

6 6 6 6 6 6 6 6 6 6

1st harm. 2nd harm. - - - - 3rd harm. - - - -

6 6 6 6 6 6 6 6

4th harm. - - - - - - - - 5th harm.

6 6 6 6 6 6 6 6

6th harm. 7th harm. 8th harm.

Harmoniques et tablature

Harmoniques et tablature (harmoniques artificielles).

```
pinchedHarmonics = {
  \textSpannerDown
  \override TextSpanner #'bound-details #'left #'text =
    \markup { \halign #-0.5 \teeny "PH" }
  \override TextSpanner #'style =
    #'dashed-line
  \override TextSpanner #'dash-period = #0.6
  \override TextSpanner #'bound-details #'right #'attach-dir = #1
  \override TextSpanner #'bound-details #'right #'text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner #'bound-details #'right #'padding = #-0.5
}
```

```
harmonics = {
  %artificial harmonics (AH)
  \textLengthOn
  <\parenthesize b b'\harmonic>4_\markup{ \teeny "AH 16" }
  <\parenthesize g g'\harmonic>4_\markup{ \teeny "AH 17" }
  <\parenthesize d' d'\harmonic>2_\markup{ \teeny "AH 19" }
  %pinched harmonics (PH)
  \pinchedHarmonics
  <a'\harmonic>2\startTextSpan
  <d'\harmonic>4
  <e'\harmonic>4\stopTextSpan
  %tapped harmonics (TH)
```

```

<\parenthesize g\4 g'\harmonic>4_\markup{ \teeny "TH 17" }
<\parenthesize a\4 a'\harmonic>4_\markup{ \teeny "TH 19" }
<\parenthesize c'\3 c''\harmonic>2_\markup{ \teeny "TH 17" }
%touch harmonics (TCH)
a4( <e''\harmonic>2. )_\markup{ \teeny "TCH" }
}

frettedStrings = {
  %artificial harmonics (AH)
  \harmonicByFret #4 g4\3
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 g2\3
  %pinched harmonics (PH)
  \harmonicByFret #7 d2\4
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 a4\5
  %tapped harmonics (TH)
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 d4\4
  \harmonicByFret #5 g2\3
  %touch harmonics (TCH)
  a4 \harmonicByFret #9 g2.\3
}

\score {
  <<
    \new Staff {
      \new Voice {
        \clef "treble_8"
        \harmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \frettedStrings
      }
    }
  >>
}

```

The image displays a musical score with two staves. The top staff is a standard musical staff with a treble clef and a common time signature (C). It contains several notes, each with a diamond-shaped marker above it. Below the notes are labels: AH 16, AH 17, AH 19, PH, TH 17, TH 19, TH 17, and TCH. The bottom staff is a tablature staff, indicated by the 'T' symbol at the beginning. It shows fret numbers in parentheses: (4), (5), (7), (7), (5), (7), (5), and 2 (9). The notes on the top staff are aligned with the fret numbers on the bottom staff.

Glissando et tablature

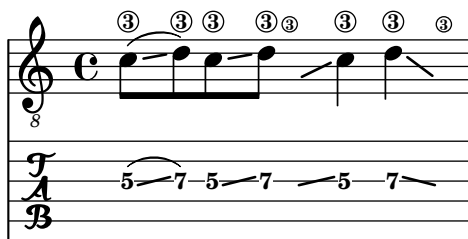
Un glissando s'indique dans un TabStaff tout comme dans un Staff.

```

slides = {
  c'8\3(\glissando d'8\3)
  c'8\3\glissando d'8\3
  \hideNotes
  \grace { g16\3\glissando }
  \unHideNotes
  c'4\3
  \afterGrace d'4\3\glissando {
    \stemDown \hideNotes
    g16\3 }
  \unHideNotes
}

\score {
  <<
    \new Staff { \clef "treble_8" \slides }
    \new TabStaff { \slides }
  >>
  \layout {
    \context {
      \Score
      \override Glissando #'minimum-length = #4
      \override Glissando #'springs-and-rods =
        #ly:spanner::set-spacing-rods
      \override Glissando #'thickness = #2
    }
  }
}

```



Glissando d'accords et tablature

Un glissando sur des accords s'indique dans un `TabStaff` de la même manière que dans un `Staff`, à ceci près que nous aurons besoin des numéros de corde afin de déterminer correctement les frets d'arrivée.

```

myMusic = \relative c' {
  <c\3 e\2 g\1>1 \glissando <f\3 a\2 c\1>
}

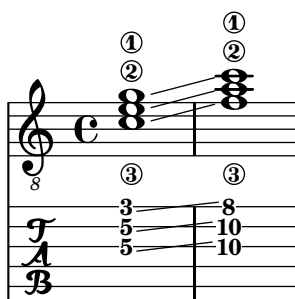
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \myMusic
    }
    \new TabStaff {

```

```

\myMusic
}
>>
}

```



Voir aussi

Manuel de notation : [Hampes], page 207, [Harmoniques], page 315, [Glissando], page 125, [Répétition d'accords], page 151, [Répétitions explicites], page 143.

Morceaux choisis : Section “Cordes frettées” dans *Morceaux choisis*.

Référence des propriétés internes : Section “TabNoteHead” dans *Référence des propriétés internes*, Section “TabStaff” dans *Référence des propriétés internes*, Section “TabVoice” dans *Référence des propriétés internes*, Section “Beam” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

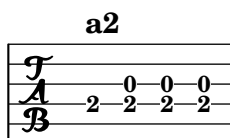
Les accords ne subissent aucun traitement particulier ; de ce fait, la sélection automatique des cordes peut attribuer une même corde pour deux notes différentes de l'accord.

Afin que `\partcombine` fonctionne avec des tablatures, on doit ajouter au contexte `TabStaff` des voix fantômes :

```

melodia = \partcombine { e4 g g g } { e4 e e e }
<<
\new TabStaff <<
  \new TabVoice = "one" s1
  \new TabVoice = "two" s1
  \new TabVoice = "shared" s1
  \new TabVoice = "solo" s1
  { \melodia }
>>
>>

```



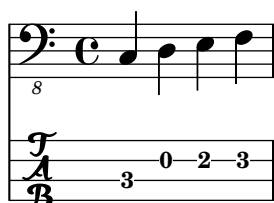
Le support des modes de jeu propres à la guitare se limite aux sons harmoniques et aux glissandos.

Tablatures personnalisées

Sous LilyPond, la case correspondant à une note jouée sur une corde donnée est calculée automatiquement. Pour ce faire, l'accordage doit être spécifié. L'accordage des cordes est donné par la propriété `stringTunings`.

LilyPond possède des accordages prédéfinis pour le banjo, la mandoline, la guitare et la guitare basse ainsi que le ukulele et les cordes d'orchestre. Lilypond calcule automatiquement la transposition correspondant à ces accordages. L'exemple ci-dessous est pour guitare basse, il sonnera donc une octave en dessous de ce qui est écrit.

```
<<
\new Voice \with {
  \override StringNumber #'stencil = ##f
} {
  \clef "bass_8"
  \relative c, {
    c4 d e f
  }
}
\new TabStaff \with {
  stringTunings = #bass-tuning
} {
  \relative c, {
    c4 d e f
  }
}
>>
```



L'accordage par défaut est `guitar-tuning`; il correspond à l'accordage standard d'une guitare : mi la ré sol si mi (EADGBE). D'autres accordages prédéfinis sont disponibles : `guitar-open-g-tuning`, `mandolin-tuning` et `banjo-open-g-tuning`. Les accordages prédéfinis sont répertoriés dans le fichier `'ly/string-tunings-init.ly'`.

LilyPond vous permet de créer n'importe quel accordage. L'accordage du contexte en cours se détermine à l'aide de la fonction `\stringTuning`. Celle-ci prend deux arguments : une représentation symbolique qui gardera l'accordage en mémoire, et une construction d'accord définissant la hauteur des différentes cordes. Les hauteurs fournies s'expriment impérativement en mode absolu – voir [Hauteurs avec octave absolue], page 1. La corde ayant le numéro le plus élevé (généralement la note la plus basse) est mentionnée en premier.

Nous pouvons ainsi définir par exemple l'accordage d'un instrument à quatre cordes accordées do sol ré la, et en anglais : `a''`, `d''`, `g'`, et `c'` :

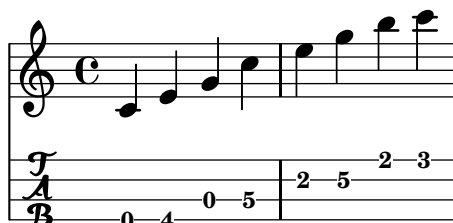
```
mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}

<<
\new Staff {
  \clef treble
  \mynotes
}
```

```

\new TabStaff {
  \#(define custom-tuning #{ \stringTuning <c' g' d'' a''> #})
  \set Staff.stringTunings = #custom-tuning
  \mynotes
}
>>

```



La propriété `stringTunings` permet aussi au `FretBoards` de calculer automatiquement les diagrammes de frets.

L'accordage fait partie de la clé permettant d'identifier les diagrammes prédéfinis – voir [\[Tablatures prédéfinies\]](#), page 344.

Nous pourrions donc écrire l'exemple précédent ainsi :

```

custom-tuning = \stringTuning <c' g' d'' a''>

```

```

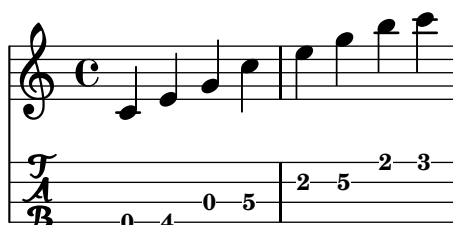
mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}

```

```

<<
\new Staff {
  \clef treble
  \mynotes
}
\new TabStaff {
  \set TabStaff.stringTunings = #custom-tuning
  \mynotes
}
>>

```



L'accordage est constitué, en interne, par une liste Scheme des hauteurs de note correspondant aux cordes à vide, une note pour chaque corde, classée par numéro de corde de 1 à n, où la corde 1 est la plus haute dans la tablature et n la plus basse. Cela revient généralement à classer les cordes de la plus aiguë à la plus grave, mais certains instruments (comme le ukulele) n'ont pas les cordes classées par hauteur.

Chaque hauteur de corde incluse dans un accordage est un objet LilyPond de type *pitch*. Les objets *pitch* sont créés par la fonction `ly:make-pitch` – voir [Section A.21 \[Fonctions Scheme\]](#), page 732.

La fonction `\stringTuning` permet de créer de tels objets à partir de la saisie d'un accord.

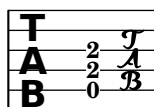
Lilypond calcule automatiquement le nombre de cordes à représenter dans la tablature (`TabStaff`) ainsi que dans le `FretBoard` en comptant le nombre d'éléments définis dans le `stringTunings`.

Les différents contextes `TabStaff` utiliseront par défaut un même accordage personnalisé dès lors que votre fichier comportera une clause

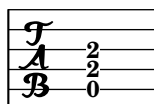
```
\layout {
  \context {
    \TabStaff
    stringTunings = \stringTuning \notemode { <c' g' d'' a''> }
  }
}
```

LilyPond dispose d'une clef de tablature moderne.

```
\new TabStaff {
  \clef moderntab
  <a, e a>1
  \break
  \clef tab
  <a, e a>1
}
```



2



Cette clef moderne prend en charge les tablatures de quatre à sept cordes.

Voir aussi

Manuel de notation : [Hauteurs avec octave absolue], page 1, [Tablatures prédéfinies], page 344, Section A.21 [Fonctions Scheme], page 732.

Fichiers d'initialisation : `'ly/string-tunings-init.ly'`, `'scm/tablature.scm'`.

Morceaux choisis : Section “Cordes frettées” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Tab_note_heads_engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Le calcul automatique des tablatures se révèle inapproprié pour les instruments dont l'ordre des cordes ne correspond pas à l'ordre des hauteurs, comme le ukulele.

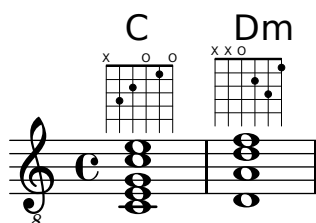
Tablatures sous forme d'étiquette

On peut ajouter des diagrammes d'accord au-dessus de n'importe quelle note, en tant qu'objets `\markup`. Ces objets contiennent toutes les informations du diagramme d'accord. Il y a pour les définir trois interfaces « diagramme d'accord » (*fret-diagram* en anglais) : *standard*, *terse* et *verbose*. Ces trois interfaces produiront des diagrammes similaires mais permettent des niveaux

de personnalisation différents. Des détails à propos des interfaces de type `\markup` peuvent être trouvés à l'annexe [Section A.10.5 \[Instrument Specific Markup\]](#), page 675.

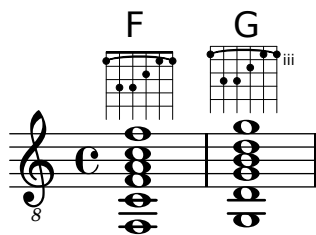
Dans l'interface standard des diagrammes d'accord, on indiquera le numéro de corde et le numéro de case pour chaque point placé sur une corde. Les cordes à vide et étouffées peuvent aussi être indiquées.

```
<<
\context ChordNames {
  \chordmode {
    c1 d:m
  }
}
\context Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram #"6-x;5-3;4-2;3-o;2-1;1-o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram #"6-x;5-x;4-o;3-2;2-3;1-1;"
  }
}
>>
```



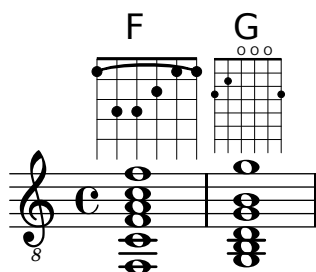
Les indications de barré peuvent aussi être ajoutées au diagramme d'accord dans l'interface standard :

```
<<
\context ChordNames {
  \chordmode {
    f1 g
  }
}
\context Staff {
  \clef "treble_8"
  <f, c f a c' f'>1^\markup {
    \fret-diagram #"c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
  }
  <g, d g b d' g'>1^\markup {
    \fret-diagram #"c:6-1-3;6-3;5-5;4-5;3-4;2-3;1-3;"
  }
}
>>
```



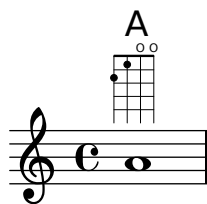
La taille du diagramme d'accord ainsi que le nombre de cases représentées peuvent aussi être modifiés dans l'interface standard.

```
<<
\context ChordNames {
  \chordmode {
    f1 g
  }
}
\context Staff {
  \clef "treble_8"
  <f, c f a c' f'>1^\markup {
    \fret-diagram #s:1.5;c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
  }
  <g, b, d g b g'>1^\markup {
    \fret-diagram #h:6;6-3;5-2;4-o;3-o;2-o;1-3;"
  }
}
>>
```



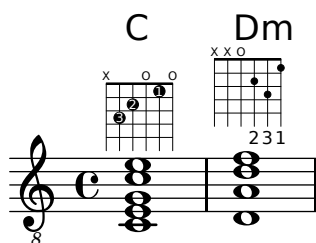
Le nombre de cordes dans les diagrammes d'accord peut être modifié dans l'interface standard pour s'adapter à différents instruments tels que le banjo et le ukulele.

```
<<
\context ChordNames {
  \chordmode {
    a1
  }
}
\context Staff {
  % An 'A' chord for ukulele
  a'1^\markup {
    \fret-diagram #w:4;4-2-2;3-1-1;2-o;1-o;"
  }
}
>>
```



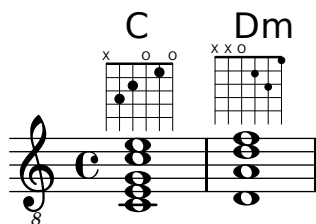
Des indications de doigtés peuvent être ajoutées, et le positionnement de ces doigtés peut être modifié dans l'interface standard.

```
<<
\context ChordNames {
  \chordmode {
    c1 d:m
  }
}
\context Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram #"f:1;6-x;5-3-3;4-2-2;3-o;2-1-1;1-o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram #"f:2;6-x;5-x;4-o;3-2-2;2-3-3;1-1-1;"
  }
}
>>
```



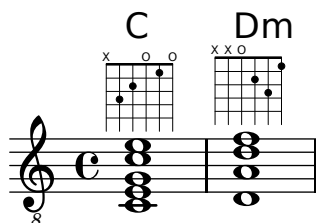
La taille ainsi que la position des points peuvent être contrôlées dans l'interface standard.

```
<<
\context ChordNames {
  \chordmode {
    c1 d:m
  }
}
\context Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram #"d:0.35;6-x;5-3;4-2;3-o;2-1;1-o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram #"p:0.2;6-x;5-x;4-o;3-2;2-3;1-1;"
  }
}
>>
```



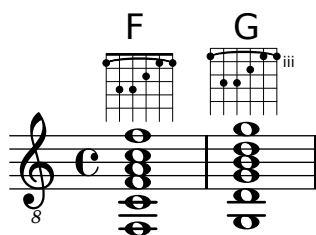
Dans l'interface `fret-diagram-terse`, les numéros de corde sont omis ; les numéros de corde sont induits par la présence de points-virgules. Il y a un point-virgule pour chaque corde du diagramme. Le premier point-virgule correspondant au plus haut numéro de corde, le dernier à la première corde. Les cordes étouffées, les cordes à vide ainsi que les numéros de case peuvent y être indiqués.

```
<<
\context ChordNames {
  \chordmode {
    c1 d:m
  }
}
\context Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram-terse #"x;3;2;o;1;o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram-terse #"x;x;o;2;3;1;"
  }
}
>>
```



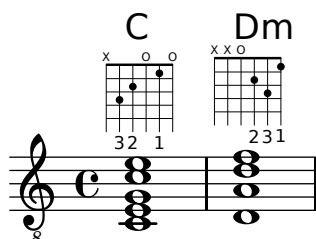
Les indications de barré peuvent être incluses dans l'interface `fret-diagram-terse`.

```
<<
\context ChordNames {
  \chordmode {
    f1 g
  }
}
\context Staff {
  \clef "treble_8"
  <f, c f a c' f'>1^\markup {
    \fret-diagram-terse #"1-(;3;3;2;1;1-);"
  }
  <g, d g b d' g'>1^\markup {
    \fret-diagram-terse #"3-(;5;5;4;3;3-);"
  }
}
>>
```



Les indications de doigtés peuvent être inclus dans l'interface `fret-diagram-terse`.

```
<<
\context ChordNames {
  \chordmode {
    c1 d:m
  }
}
\context Staff {
  \override Voice.TextScript
    #'(fret-diagram-details finger-code) = #'below-string
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram-terse #"x;3-3;2-2;o;1-1;o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram-terse #"x;x;o;2-2;3-3;1-1;"
  }
}
>>
```



Les autres propriétés des diagrammes d'accord doivent être ajustées en utilisant la commande `\override` dans l'interface `fret-diagram-terse`.

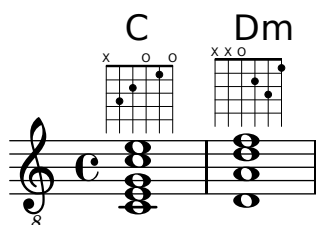
L'interface `fret-diagram-verbose` est au format d'une liste Scheme. Chaque élément de la liste décrit un objet devant être placé dans le diagramme d'accord.

```
<<
\context ChordNames {
  \chordmode {
    c1 d:m
  }
}
\context Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram-verbose #'(
      (mute 6)
      (place-fret 5 3)
      (place-fret 4 2)
      (open 3)
      (place-fret 2 1)
    )
  }
}
>>
```

```

        (open 1)
    )
}
<d a d' f'>1^\markup {
  \fret-diagram-verbose #'(
    (mute 6)
    (mute 5)
    (open 4)
    (place-fret 3 2)
    (place-fret 2 3)
    (place-fret 1 1)
  )
}
}
>>

```



Les indications de doigté et de barré peuvent être décrits dans l'interface **fret-diagram-verbose**. Particularité propre à l'interface **fret-diagram-verbose** : l'indication de capodastre dans le diagramme d'accord. L'indication de capodastre est une petite ligne transversale aux cordes. La case avec le capodastre est la case la plus basse du diagramme d'accord.

```

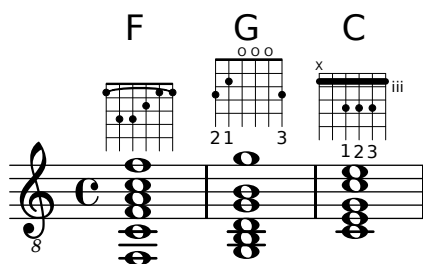
<<
  \context ChordNames {
    \chordmode {
      f1 g c
    }
  }
  \context Staff {
    \clef "treble_8"
    \override Voice.TextScript
      #'(fret-diagram-details finger-code) = #'below-string
    <f, c f a c' f'>1^\markup {
      \fret-diagram-verbose #'(
        (place-fret 6 1)
        (place-fret 5 3)
        (place-fret 4 3)
        (place-fret 3 2)
        (place-fret 2 1)
        (place-fret 1 1)
        (barre 6 1 1)
      )
    }
    <g, b, d g b g'>1^\markup {
      \fret-diagram-verbose #'(
        (place-fret 6 3 2)
        (place-fret 5 2 1)
      )
    }
  }
}

```

```

        (open 4)
        (open 3)
        (open 2)
        (place-fret 1 3 3)
    )
}
<c e g c' e'>1^\markup {
  \fret-diagram-verbose #'(
    (capo 3)
    (mute 6)
    (place-fret 4 5 1)
    (place-fret 3 5 2)
    (place-fret 2 5 3)
  )
}
}
>>

```



Toutes les autres propriétés du diagramme d'accord doivent être indiquées en utilisant la commande `\override` lorsque l'on utilise l'interface `fret-diagram-verbose`.

La disposition graphique d'un diagramme d'accord peut être modifiée suivant les préférences de l'utilisateur grâce aux propriétés de l'interface `fret-diagram-interface`. Des détails se trouvent dans [Section “fret-diagram-interface”](#) dans *Référence des propriétés internes*. Pour un diagramme d'accord, les propriétés de l'interface dépendent de `Voice.TextScript`.

Morceaux choisis

Orientation des diagrammes de fret

Les diagrammes de fret peuvent s'orienter de trois manières différentes. Ils s'aligneront par défaut sur la corde du haut ou le sommet du fret.

```
\include "predefined-guitar-fretboards.ly"
```

```

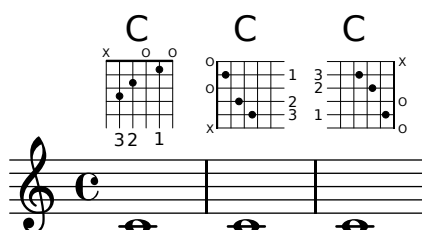
<<
\chords {
  c1
  c1
  c1
}
\new FretBoards {
  \chordmode {
    c1
    \override FretBoard #'(fret-diagram-details orientation) =
      #'landscape
    c1
  }
}

```

```

\override FretBoard #'(fret-diagram-details orientation) =
  #'opposing-landscape
c1
}
}
\new Voice {
  c'1
  c'1
  c'
}
>>

```



Personnalisation des diagrammes de fret

Les propriétés d'un diagramme de fret sont modifiables grâce au '`fret-diagram-details`'. Lorsqu'ils sont générés sous forme de `\markup`, rien n'empêche de modifier les diagrammes en jouant sur les réglages de l'objet `Voice.TextScript` ou bien directement sur le *markup*.

```

<<
\chords { c1 | c | c | d }

\new Voice = "mel" {
  \textLengthOn
  % Set global properties of fret diagram
  \override TextScript #'size = #'1.2
  \override TextScript
    #'(fret-diagram-details finger-code) = #'in-dot
  \override TextScript
    #'(fret-diagram-details dot-color) = #'white

  %% C major for guitar, no barre, using defaults
  % terse style
  c'1~\markup { \fret-diagram-terse #"x;3-3;2-2;o;1-1;o;" }

  %% C major for guitar, barred on third fret
  % verbose style
  % size 1.0
  % roman fret label, finger labels below string, straight barre
  c'1~\markup {
    % standard size
    \override #'(size . 1.0) {
      \override #'(fret-diagram-details . (
        (number-type . roman-lower)
        (finger-code . in-dot)
        (barre-type . straight))) {
        \fret-diagram-verbose #'((mute 6)

```



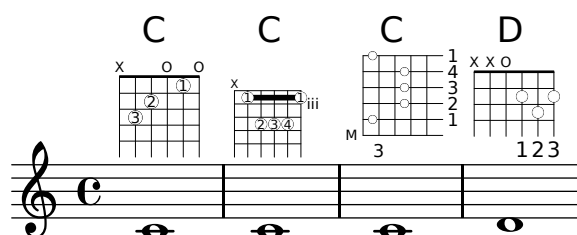
```

        (place-fret 5 3 1)
        (place-fret 4 5 2)
        (place-fret 3 5 3)
        (place-fret 2 5 4)
        (place-fret 1 3 1)
        (barre 5 1 3))
    }
}
}

%% C major for guitar, barred on third fret
% verbose style
% landscape orientation, arabic numbers, M for mute string
% no barre, fret label down or left, small mute label font
c'1^\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (number-type . arabic)
    (label-dir . -1)
    (mute-string . "M")
    (orientation . landscape)
    (barre-type . none)
    (xo-font-magnification . 0.4)
    (xo-padding . 0.3))) {
    \fret-diagram-verbose #'((mute 6)
      (place-fret 5 3 1)
      (place-fret 4 5 2)
      (place-fret 3 5 3)
      (place-fret 2 5 4)
      (place-fret 1 3 1)
      (barre 5 1 3))
    }
  }
}

%% simple D chord
% terse style
% larger dots, centered dots, fewer frets
% label below string
d'1^\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (dot-radius . 0.35)
    (dot-position . 0.5)
    (fret-count . 3))) {
    \fret-diagram-terse #"x;x;o;2-1;3-2;2-3;"
  }
}
}
}
>>

```



Voir aussi

Manuel de notation : [Section A.10.5 \[Instrument Specific Markup\]](#), page 675.

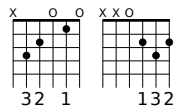
Morceaux choisis : [Section “Cordes frettées”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “fret-diagram-interface”](#) dans *Référence des propriétés internes*.

Tablatures prédéfinies

Les diagrammes d'accord peuvent être affichés en utilisant le contexte **FretBoards**. Par défaut le contexte **FretBoards** affichera des diagrammes d'accord stockés dans une table de correspondance :

```
\include "predefined-guitar-fretboards.ly"
\context FretBoards {
  \chordmode {
    c1 d
  }
}
```



Les diagrammes d'accord définis par défaut sont dans le fichier ‘predefined-guitar-fretboards.ly’. Les diagrammes d'accord sont stockés en fonction des notes de l'accord ainsi que de l'accordage (**stringTunings**) utilisé. Le fichier d'initialisation ‘predefined-guitar-fretboards.ly’ contient les diagrammes d'accord prédéfinis uniquement pour l'accordage standard (**guitar-tuning**). Des diagrammes d'accords peuvent être définis pour d'autres instruments ou d'autres accordages en suivant les exemples du fichier ‘predefined-guitar-fretboards.ly’.

Les diagrammes de fret propres au ukulele se trouvent dans le fichier ‘predefined-ukulele-fretboards.ly’.

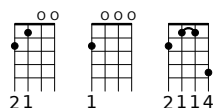
```
\include "predefined-ukulele-fretboards.ly"
```

```
myChords = \chordmode { a1 a:m a:aug }
```

```
\new ChordNames {
  \myChords
}
```

```
\new FretBoards {
  \set Staff.stringTunings = #ukulele-tuning
  \myChords
}
```

A Am A+



Les diagrammes de fret propres à la mandoline se trouvent dans le fichier ‘predefined-mandolin-fretboards.ly’.

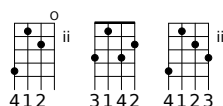
```
\include "predefined-mandolin-fretboards.ly"
```

```
myChords = \chordmode { c1 c:m7.5- c:aug }
```

```
\new ChordNames {
  \myChords
}
```

```
\new FretBoards {
  \set Staff.stringTunings = #mandolin-tuning
  \myChords
}
```

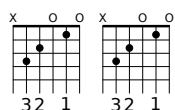
C C[∅] C+



Les notes des accords peuvent être entrées aussi bien comme musique simultanée qu’en utilisant le mode accord (voir [Généralités sur le mode accords](#), page 385).

```
\include "predefined-guitar-fretboards.ly"
```

```
\context FretBoards {
  \chordmode {c1}
  <c' e' g'>1
}
```

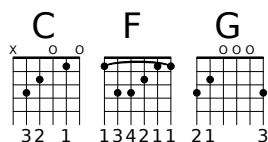


Il est courant d’afficher simultanément le nom des accords et les diagrammes d’accord correspondants. Ceci s’obtient en mettant en parallèle un contexte **ChordNames** et un contexte **FretBoards**, tout en affectant aux deux la même musique.

```
\include "predefined-guitar-fretboards.ly"
```

```
mychords = \chordmode{
  c1 f g
}
```

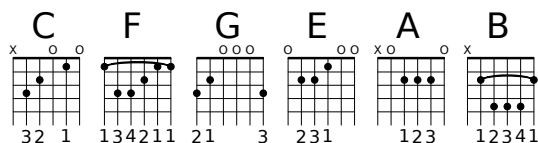
```
<<
  \context ChordNames {
    \mychords
  }
  \context FretBoards {
    \mychords
  }
>>
```



Les diagrammes d'accord prédéfinis sont transposables tant qu'il y a un diagramme correspondant à l'accord transposé dans la base des diagrammes d'accord.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 f g
}
```

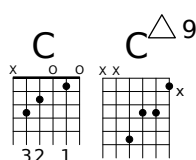
```
mychordlist = {
  \mychords
  \transpose c e { \mychords }
}
<<
  \context ChordNames {
    \mychordlist
  }
  \context FretBoards {
    \mychordlist
  }
>>
```



La table des diagrammes d'accord contient sept types d'accord (majeur, mineur, augmenté, diminué, septième de dominante, majeur sept, mineur sept) pour chacune des 17 fondamentales possibles. Une liste complète des diagrammes d'accords prédéfinis se trouve à l'annexe [Tablatures prédéfinies], page 344. S'il n'y a pas d'entrée dans la table pour un accord donné, le graveur `Fretboard_engraver` calculera le diagramme d'accord en utilisant la fonctionnalité automatique décrite dans [Tablatures automatiques], page 354.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 c:maj9
}
```

```
<<
  \context ChordNames {
    \mychords
  }
  \context FretBoards {
    \mychords
  }
>>
```



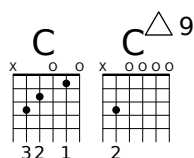
Des diagrammes d'accord peuvent être ajoutés à la table des diagrammes d'accord. Pour ajouter un diagramme d'accord, il faut spécifier l'accord correspondant au diagramme, l'accord utilisé et la définition du diagramme. Cette définition de diagramme peut être aussi bien de type *terse* que *verbose*.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
                        \chordmode { c:maj9 }
                        #guitar-tuning
                        #"x;3-2;o;o;o;o;"

mychords = \chordmode {
  c1 c:maj9
}

<<
  \context ChordNames {
    \mychords
  }
  \context FretBoards {
    \mychords
  }
>>
```



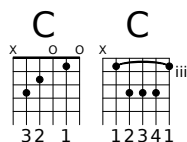
On peut enregistrer différents diagrammes pour un même accord en les définissant à des octaves différentes. Notez qu'il faut un intervalle de deux octaves, le premier servant à la transposition.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
                        \chordmode { c'' }
                        #guitar-tuning
                        #(offset-fret 2 (chord-shape 'bes guitar-tuning))

mychords = \chordmode {
  c1 c''
}

<<
  \context ChordNames {
    \mychords
  }
  \context FretBoards {
    \mychords
  }
>>
```



En plus des diagrammes d'accord, LilyPond possède une liste interne de carrures d'accord. Les carrures d'accords sont des diagrammes d'accord qui peuvent être transposés le long du manche. Les carrures d'accords peuvent être ajoutées à la liste interne et être ensuite utilisées pour définir des accords prédéfinis. Puisqu'elles sont transposables le long du manche, les carrures d'accord ne contiennent généralement pas de corde à vide. Tout comme les diagrammes d'accord, les carrures d'accord sont définies grâce aux interfaces `fret-diagram-terse` ou `fret-diagram-verbose`.

```
\include "predefined-guitar-fretboards.ly"

% Add a new chord shape

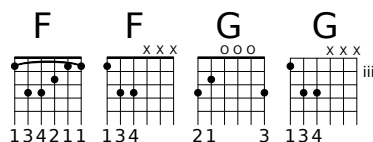
\addChordShape #'powerf #guitar-tuning #"1-1;3-3;3-4;x;x;x;"

% add some new chords based on the power chord shape

\storePredefinedDiagram #default-fret-table
    \chordmode { f'' }
    #guitar-tuning
    #(chord-shape 'powerf guitar-tuning)
\storePredefinedDiagram #default-fret-table
    \chordmode { g'' }
    #guitar-tuning
    #(offset-fret 2 (chord-shape 'powerf guitar-tuning))

mychords = \chordmode{
  f1 f'' g g''
}

<<
  \context ChordNames {
    \mychords
  }
  \context FretBoards {
    \mychords
  }
>>
```



La disposition graphique d'un diagramme d'accord peut être modifiée suivant les préférences de l'utilisateur grâce aux propriétés de l'interface `fret-diagram-interface`. Pour plus d'information, consultez [Section "fret-diagram-interface"](#) dans *Référence des propriétés internes*. Pour un diagramme d'accord donné, les propriétés de l'interface dépendent de `FretBoards.FretBoard`.

Morceaux choisis

Personnalisation de diagrammes de fret

Les propriétés d'un diagramme de fret sont définies par les 'fret-diagram-details. En matière de diagramme de fret, les adaptations s'appliquent à l'objet `FretBoards.FretBoard`. Un `FretBoards` est comparable à un `Voice` : il s'agit d'un contexte du plus bas niveau, et il n'est donc pas primordial de l'instancier de manière explicite pour adapter ses propriétés.

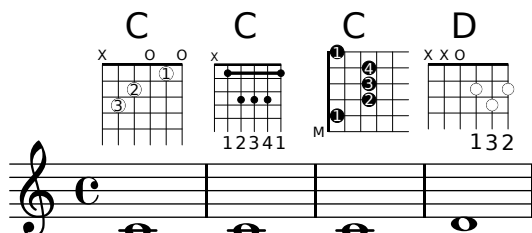
```
\include "predefined-guitar-fretboards.ly"
\storePredefinedDiagram #default-fret-table \chordmode { c' }
      #guitar-tuning
      #"x;1-1-(;3-2;3-3;3-4;1-1-);"

<<
\new ChordNames {
  \chordmode { c1 | c | c | d }
}
\new FretBoards {
  % Set global properties of fret diagram
  \override FretBoards.FretBoard #'size = #'1.2
  \override FretBoard
    #'(fret-diagram-details finger-code) = #'in-dot
  \override FretBoard
    #'(fret-diagram-details dot-color) = #'white
  \chordmode {
    c
    \once \override FretBoard #'size = #'1.0
    \once \override FretBoard
      #'(fret-diagram-details barre-type) = #'straight
    \once \override FretBoard
      #'(fret-diagram-details dot-color) = #'black
    \once \override FretBoard
      #'(fret-diagram-details finger-code) = #'below-string
    c'
    \once \override FretBoard
      #'(fret-diagram-details barre-type) = #'none
    \once \override FretBoard
      #'(fret-diagram-details number-type) = #'arabic
    \once \override FretBoard
      #'(fret-diagram-details orientation) = #'landscape
    \once \override FretBoard
      #'(fret-diagram-details mute-string) = #'M"
    \once \override FretBoard
      #'(fret-diagram-details label-dir) = #LEFT
    \once \override FretBoard
      #'(fret-diagram-details dot-color) = #'black
    c'
    \once \override FretBoard
      #'(fret-diagram-details finger-code) = #'below-string
    \once \override FretBoard
      #'(fret-diagram-details dot-radius) = #0.35
    \once \override FretBoard
      #'(fret-diagram-details dot-position) = #0.5
    \once \override FretBoard
```

```

      #'(fret-diagram-details fret-count) = #3
    d
  }
}
\new Voice {
  c'1 | c' | c' | d'
}
>>

```



Création de diagrammes de fret prédéfinis pour d'autres instruments

La liste des diagrammes standards prédéfinis pour la guitare peut être augmentée d'autres définitions spécifiques à d'autres instruments. Voici comment définir un nouvel accordage ainsi que quelques diagrammes prédéfinis pour le « cuatro vénézuélien ».

Cet exemple illustre aussi la manière d'ajouter des doigtés aux accords ; ils serviront de référence pour la boucle d'accord et seront indiqués dans les diagrammes et le **TabStaff**, mais pas dans la musique.

Ces diagrammes ne peuvent pas être transposés, dans la mesure où ils contiennent des informations sur les cordes. Ceci est amené à évoluer.

```

% add FretBoards for the Cuatro
% Note: This section could be put into a separate file
% predefined-cuatro-fretboards.ly
% and \included into each of your compositions

```

```

cuatroTuning = #`((ly:make-pitch 0 6 0)
                  ,(ly:make-pitch 1 3 SHARP)
                  ,(ly:make-pitch 1 1 0)
                  ,(ly:make-pitch 0 5 0))

```

```

dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }
aMajSeven = { <a\4 cis\1 e\3 g\2> }
dMajSeven = { <a\4 c\1 d\3 fis\2> }
gMajor = { <b\4 b\1 d\3 g\2> }

```

```

\storePredefinedDiagram #default-fret-table \dSix
                        #cuatroTuning
                        #"o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
                        #cuatroTuning
                        #"o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
                        #cuatroTuning
                        #"o;2-2;1-1;2-3;"
\storePredefinedDiagram #default-fret-table \dMajSeven

```



```

        #cuatroTuning
        #"o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
        #cuatroTuning
        #"2-2;o;1-1;o;"

% end of potential include file /predefined-cuatro-fretboards.ly

#(set-global-staff-size 16)

primerosNames = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }

    \new FretBoards {
      \set Staff.stringTunings = #cuatroTuning
%      \override FretBoard
%      #'(fret-diagram-details string-count) = #'4
      \override FretBoard
        #'(fret-diagram-details finger-code) = #'in-dot
      \primeros
    }

    \new TabStaff \relative c'' {
      \set TabStaff.stringTunings = #cuatroTuning
      \primeros
    }
  >>

```

```

\layout {
  \context {
    \Score
    \override SpacingSpanner
      #'base-shortest-duration = #(ly:make-moment 1 16)
  }
}
\midi { }
}

```

Changement d'accord et diagramme de fret

Vous pouvez opter pour n'imprimer les diagrammes de fret qu'à l'occasion d'un changement d'accord ou de saut de ligne.

```
\include "predefined-guitar-fretboards.ly"
```

```

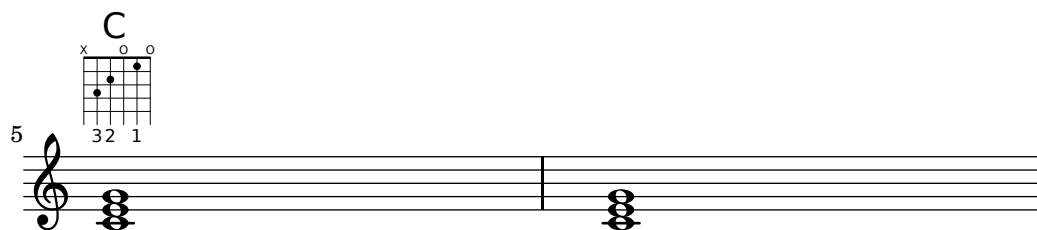
myChords = \chordmode {
  c1 c1 \break
  \set chordChanges = ##t
  c1 c1 \break
  c1 c1
}

```

```

<<
  \new ChordNames { \myChords }
  \new FretBoards { \myChords }
  \new Staff { \myChords }
>>

```



Diagrammes de fret alternatifs

Vous pouvez tout à fait créer des tables de diagrammes de fret supplémentaires, notamment dans l'optique d'un fret alternatif pour un accord donné.

Avant de pouvoir utiliser un diagramme alternatif, vous devrez alimenter une table à cet effet. Les différents diagrammes seront ajoutés à cette table.

Il peut aussi bien s'agir d'une table vide, que de la recopie d'une table existante.

La table servant de base pour les diagrammes prédéfinis est sélectionnée par la propriété `\predefinedDiagramTable`.

```
\include "predefined-guitar-fretboards.ly"
```

```
% Make a blank new fretboard table
```

```
 #(define custom-fretboard-table-one (make-fretboard-table))
```

```
% Make a new fretboard table as a copy of default-fret-table
```

```
 #(define custom-fretboard-table-two (make-fretboard-table default-fret-table))
```

```
% Add a chord to custom-fretboard-table-one
```

```
 \storePredefinedDiagram #custom-fretboard-table-one
    \chordmode{c}
    #guitar-tuning
    "3-(;3;5;5;5;3-);"
```

```
% Add a chord to custom-fretboard-table-two
```

```
 \storePredefinedDiagram #custom-fretboard-table-two
    \chordmode{c}
    #guitar-tuning
    "x;3;5;5;5;o;"
```

```
<<
```

```
 \chords {
   c1 | d1 |
   c1 | d1 |
   c1 | d1 |
 }
```

```
 \new FretBoards {
```

```
   \chordmode {
     \set predefinedDiagramTable = #default-fret-table
     c1 | d1 |
     \set predefinedDiagramTable = #custom-fretboard-table-one
     c1 | d1 |
     \set predefinedDiagramTable = #custom-fretboard-table-two
     c1 | d1 |
   }
```

```
}
```

```
 \new Staff {
```

```
   \clef "treble_8"
```

```

<<
  \chordmode {
    c1 | d1 |
    c1 | d1 |
    c1 | d1 |
  }
  {
    s1\_markup "Default table" | s1 |
    s1\_markup \column {"New table" "from empty"} | s1 |
    s1\_markup \column {"New table" "from default"} | s1 |
  }
>>
}
>>

```

Diagram illustrating the output of the notation, showing six chords (C, D, C, D, C, D) with their corresponding fretboard diagrams and musical notation. The diagrams are labeled: Default table, New table from empty, and New table from default.

Voir aussi

Manuel de notation : [Tablatures personnalisées], page 331, [Tablatures automatiques], page 354, [Généralités sur le mode accords], page 385, [Tablatures prédéfinies], page 344.

Fichiers d'initialisation : 'ly/predefined-guitar-fretboards.ly', 'ly/predefined-guitar-ninth-fretboards.ly', 'ly/predefined-ukulele-fretboards.ly', 'ly/predefined-mandolin-fretboards.ly'.

Morceaux choisis : Section “Cordes frettées” dans *Morceaux choisis*.

Référence des propriétés internes : Section “fret-diagram-interface” dans *Référence des propriétés internes*.

Tablatures automatiques

Les diagrammes d'accord peuvent être créés automatiquement ; il suffit d'affecter les notes à un contexte `FretBoards`. Si aucun diagramme prédéfini n'est disponible pour les notes entrées avec l'accordage actuel (`stringTunings`), les cordes et cases correspondant aux notes seront automatiquement calculées.

```

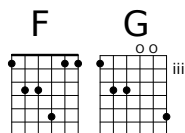
<<
  \context ChordNames {
    \chordmode {
      f1 g
    }
  }
  \context FretBoards {
    <f, c f a c' f'>1
    <g,\6 b, d g b g'>1
  }
  \context Staff {

```

```

\clef "treble_8"
<f, c f a c' f'>1
<g, b, d g b' g'>1
}
>>

```

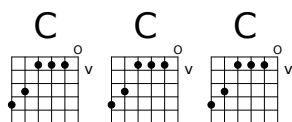


Dans la mesure où aucun diagramme prédéfini n'est chargé par défaut, le calcul automatique des diagrammes d'accord est le comportement par défaut. Dès que les diagrammes par défaut sont chargés, le calcul automatique peut être activé ou désactivé par des commandes prédéfinies :

```

\storePredefinedDiagram #default-fret-table
    <c e g c' e'>
    #guitar-tuning
    #"x;3-1-(;5-2;5-3;5-4;3-1-1-);"
<<
\context ChordNames {
  \chordmode {
    c1 c c
  }
}
\context FretBoards {
  <c e g c' e'>1
  \predefinedFretboardsOff
  <c e g c' e'>1
  \predefinedFretboardsOn
  <c e g c' e'>1
}
\context Staff {
  \clef "treble_8"
  <c e g c' e'>1
  <c e g c' e'>1
  <c e g c' e'>1
}
>>

```



Le calculateur se trouvera parfois incapable de trouver un diagramme d'accord convenable. On peut souvent y remédier en assignant les notes aux cordes. Dans bien des cas, il suffit de placer manuellement une seule note pour que les autres soient alors placées de manière appropriée par le contexte `FretBoards`.

Il est possible d'ajouter des indications de doigté aux diagrammes de fret.

```

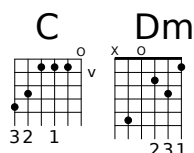
<<
\context ChordNames {
  \chordmode {

```

```

    c1 d:m
  }
}
\context FretBoards {
  <c-3 e-2 g c'-1 e'>1
  <d a-2 d'-3 f'-1>1
}
\context Staff {
  \clef "treble_8"
  <c e g c' e'>1
  <d a d' f'>1
}
>>

```

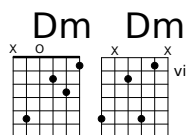


La propriété `minimumFret` permet de définir la case minimale qui servira à calculer les cordes et les cases du contexte `FretBoard`.

```

<<
\context ChordNames {
  \chordmode {
    d1:m d:m
  }
}
\context FretBoards {
  <d a d' f'>1
  \set FretBoards.minimumFret = #5
  <d a d' f'>1
}
\context Staff {
  \clef "treble_8"
  <d a d' f'>1
  <d a d' f'>1
}
>>

```



Les cordes et les cases du contexte `FretBoards` sont liées à la propriété `stringTunings`, qui a là même signification que dans le contexte `TabStaff`. Voir [\[Tablatures personnalisées\]](#), page 331 pour plus d'information sur la propriété `stringTunings`.

La disposition graphique d'un diagramme d'accord peut être modifiée suivant les préférences de l'utilisateur au travers des propriétés de l'interface `fret-diagram-interface`. Pour un diagramme d'accord `FretBoards` donné, les propriétés de l'interface dépendent de `FretBoards.FretBoard`.

Commandes prédéfinies

`\predefinedFretboardsOff`, `\predefinedFretboardsOn`.

Voir aussi

Manuel de notation : [Tablatures personnalisées], page 331.

Morceaux choisis : Section “Cordes frettées” dans *Morceaux choisis*.

Référence des propriétés internes : Section “fret-diagram-interface” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Le calcul automatique des diagrammes de fret se révèle inapproprié pour les instruments dont l'ordre des cordes ne correspond pas à l'ordre des hauteurs.

Doigtés pour la main droite

Les doigtés de main droite *p-i-m-a* doivent être entrés à l'aide de l'instruction `\rightHandFinger` suivie d'un nombre.

Note : Lorsque vous utilisez l'instruction `\rightHandFinger` dans un accord, une espace **doit** précéder la fermeture de la construction par un `>`.

```
\clef "treble_8"
c4\rightHandFinger #1
e\rightHandFinger #2
g\rightHandFinger #3
c\rightHandFinger #4
<c,\rightHandFinger #1 e\rightHandFinger #2
g\rightHandFinger #3 c\rightHandFinger #4 >1
```



Pour plus de clarté, vous pouvez traduire ou abréger la commande `\rightHandFinger`, par exemple en `doigtMainDroite` ou même `MD` :

`MD = #rightHandFinger`

Morceaux choisis

Positionnement des doigtés main droite

Le positionnement des doigtés main droite, grâce à une propriété spécifique, peut se contrôler finement, comme l'indique l'exemple suivant. N'oubliez pas la construction de type accord.

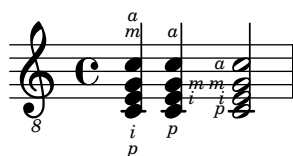
```
#(define RH rightHandFinger)
```

```
\relative c {
  \clef "treble_8"

  \set strokeFingerOrientations = #'(up down)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >4

  \set strokeFingerOrientations = #'(up right down)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >4
```

```
\set strokeFingerOrientations = #'(left)
<c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >2
}
```

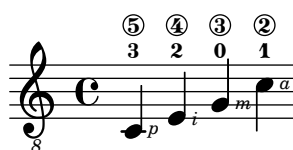


Doigtés, indications de corde, et doigtés main droite

L'exemple suivant illustre comment combiner des doigtés pour la main gauche, des indications de corde et des doigtés pour la main droite.

```
#(define RH rightHandFinger)
```

```
\relative c {
  \clef "treble_8"
  <c-3\5-\RH #1 >4
  <e-2\4-\RH #2 >4
  <g-0\3-\RH #3 >4
  <c-1\2-\RH #4 >4
}
```



Voir aussi

Morceaux choisis : [Section “Cordes frettées”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “StrokeFinger”](#) dans *Référence des propriétés internes*.

2.4.2 Guitare

La plupart des aspects en matière de notation pour guitare sont traités dans la partie commune aux instruments frettés. Il subsiste cependant quelques particularités que nous allons maintenant examiner.

Parfois l'utilisateur aimerait créer des documents de type recueil de chansons, où l'on ne trouve que des accords au-dessus des paroles. Dans la mesure où LilyPond est un éditeur de partitions, il n'est pas l'outil optimal pour des documents sans partition. Une meilleure alternative serait de recourir à un traitement de texte, un éditeur de texte ou, pour les utilisateurs expérimentés, un logiciel typographique tel que GuitarTex.

Indication de la position et du barré

Cet exemple montre comment indiquer les positions et les barrés :

```
\clef "treble_8"
b16 d g b e
\textSpannerDown
\override TextSpanner #'(bound-details left text) = #"XII "
g16\startTextSpan
```



```
b16 e g e b g\stopTextSpan
e16 b g d
```



Voir aussi

Manuel de notation : [Indication textuelle avec extension], page 214.

Morceaux choisis : Section “Cordes frettées” dans *Morceaux choisis*, Section “Signes d’interprétation” dans *Morceaux choisis*.

Indication des harmoniques et notes étouffées

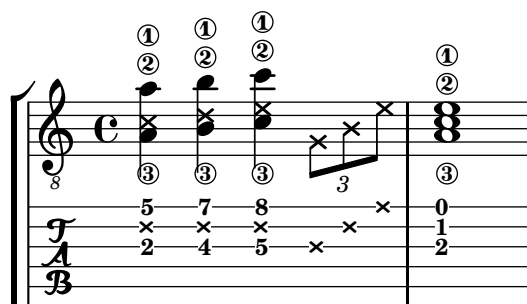
Des têtes de note spéciales peuvent servir à indiquer les notes étouffées et les sons harmoniques. Les sons harmoniques sont souvent détaillés grâce à des indications textuelles.

```
\relative c' {
  \clef "treble_8"
  \override Staff.NoteHead #'style = #'harmonic-mixed
  d~\markup { \italic { \fontsize #-2 { "harm. 12" }}} <g b>1
}
```



Les notes étouffées, ou *notes fantômes*, se rencontrent aussi bien sur une portée normale que dans une tablature :

```
music = \relative c' {
  < a\3 \deadNote c\2 a'\1 >4
  < b\3 \deadNote d\2 b'\1 >
  < c\3 \deadNote e\2 c'\1 >
  \deadNotesOn
  \times 2/3 { g8 b e }
  \deadNotesOff
  < a,\3 c\2 e\1 >1
}
\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \music
  }
  \new TabStaff {
    \music
  }
>>
```



Le *palm mute*, appelé aussi parfois *chop*, est une technique de jeu pour la guitare électrique ; elle est connue sous le nom de pizzicato par les joueurs de guitare classique. Elle consiste à poser la main droite sur les cordes juste au-dessus du chevalet, de façon à étouffer plus ou moins légèrement les notes. LilyPond permet d'indiquer ce style de jeu en affectant un profil spécifique aux têtes de note.

```
\new Voice { % Warning: explicit Voice instantiation is
              % required to have palmMuteOff work properly
              % when palmMuteOn comes at the beginning of
              % the piece.
\relative c, {
  \clef "G_8"
  \palmMuteOn
  e8~\markup { \musicglyph #"noteheads.u2do" = palm mute }
  < e b' e > e
  \palmMuteOff
  e e \palmMute e e e |
  e8 \palmMute { e e e } e e e e |
  < \palmMute e b' e >8 \palmMute { e e e } < \palmMute e b' e >2
}
}
```



Voir aussi

Manuel de notation : [Têtes de note spécifiques], page 33, Section 1.1.4 [Têtes de note], page 33.

Morceaux choisis : Section “Cordes frettées” dans *Morceaux choisis*.

Indication de *power chord*

Les *power chords* – terme anglais signifiant littéralement « accords de puissance » – s'indiquent aussi bien en mode accord que dans une construction en accord :

```
ChordsAndSymbols = {
  \chordmode {
    \powerChords
    e,,1:1.5
    a,,1:1.5.8
    \set minimumFret = #8
    c,1:1.5
    f,1:1.5.8
  }
}
```

```

\set minimumFret = #5
<a, e>1
<g d' g'>1
}
\score {
  <<
    \new ChordNames {
      \ChordsAndSymbols
    }
    \new Staff {
      \clef "treble_8"
      \ChordsAndSymbols
    }
    \new TabStaff {
      \ChordsAndSymbols
    }
  >>
}

```

The image displays a musical score for a guitar. The top staff is a treble clef staff with a common time signature (C). It contains six measures, each with a power chord: E⁵, A⁵, C⁵, F⁵, A⁵, and G⁵. Below the staff is a guitar fretboard diagram with six strings (E, A, D, G, B, E) and six frets. The fret numbers for each measure are: Measure 1 (E5): 8 on E, 8 on A; Measure 2 (A5): 2 on A, 0 on D; Measure 3 (C5): 0 on C, 8 on G; Measure 4 (F5): 10 on F, 10 on B; Measure 5 (A5): 7 on A, 5 on D; Measure 6 (G5): 8 on G, 5 on B.

Le symbole de *power chord* est désactivé dès lors que survient un accord traditionnel :

```

mixedChords = \chordmode {
  c,1
  \powerChords
  b,,1:1.5
  fis,,1:1.5.8
  g,,1:m
}
\score {
  <<
    \new ChordNames {
      \mixedChords
    }
    \new Staff {
      \clef "treble_8"
      \mixedChords
    }
    \new TabStaff {
      \mixedChords
    }
  >>
}

```

Voir aussi

Glossaire musicologique : [Section “power chord”](#) dans *Glossaire*.

Manuel de notation : [\[Extension et altération d’accords\]](#), page 388, [\[Impression des noms d’accord\]](#), page 391.

Morceaux choisis : [Section “Cordes frettées”](#) dans *Morceaux choisis*.

2.4.3 Banjo

Tablatures pour banjo

LilyPond permet d’écrire des tablatures de base pour le banjo à cinq cordes. Pour ce faire, pensez à utiliser le format de tablature pour banjo, afin d’avoir le bon nombre de cordes et le bon accordage :

```
\new TabStaff <<
  \set TabStaff.tablatureFormat = #fret-number-tablature-format-banjo
  \set TabStaff.stringTunings = #banjo-open-g-tuning
  {
    \stemDown
    g8 d' g'\5 a b g e d' |
    g4 d''8\5 b' a'\2 g'\5 e'\2 d' |
    g4
  }
>>
```

LilyPond prend en charge un certain nombre d’accordages courants pour banjo : `banjo-c-tuning` sol do sol si ré (gCGBD), `banjo-modal-tuning` sol ré sol do ré (gDGCD), `banjo-open-d-tuning` la ré fa# la ré (aDF#AD) et `banjo-open-dm-tuning` la ré fa la ré (aDFAD).

Ces accordages peuvent être convertis pour banjo à quatre cordes au moyen de la fonction `four-string-banjo` :

```
\set TabStaff.stringTunings = #(four-string-banjo banjo-c-tuning)
```

Voir aussi

Fichiers d’initialisation : `‘ly/string-tunings-init.ly’`.

Morceaux choisis : [Section “Cordes frettées”](#) dans *Morceaux choisis*.

2.5 Percussions

2.5.1 Vue d’ensemble des percussions

La notation rythmique sert avant tout aux parties de percussions ou de batterie, mais on peut aussi s’en servir à des fins pédagogiques, pour montrer le rythme d’une mélodie.

Références en matière de notation pour percussions

- Certains instruments à percussion se notent sur une portée rythmique. Vous trouverez des informations détaillées à ce sujet aux rubriques [Gravure de lignes rythmiques], page 73 et [Initialisation de nouvelles portées], page 171.
- Le rendu MIDI des percussions fait l’objet d’une rubrique dédiée : Section 3.5.6 [MIDI et percussions], page 493.

Voir aussi

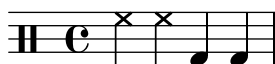
Manuel de notation : [Gravure de lignes rythmiques], page 73, [Initialisation de nouvelles portées], page 171, Section 3.5.6 [MIDI et percussions], page 493.

Morceaux choisis : Section “Percussion” dans *Morceaux choisis*.

Notation de base pour percussions

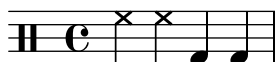
Les parties de percussions peuvent être saisies avec le mode `\drummode`, qui est l’équivalent du mode standard utilisé pour les notes à hauteur déterminée. Le moyen plus simple pour saisir une partie de percussion est d’utiliser la commande `\drums`, qui crée un contexte spécifique :

```
\drums {
  hihat4 hh bassdrum bd
}
```



Il s’agit en fait d’un raccourci pour

```
\new DrumStaff {
  \drummode {
    hihat4 hh bassdrum bd
  }
}
```



Chaque instrument de percussion peut avoir, dans le fichier LilyPond, un nom complet et un nom raccourci. Ces noms sont inventoriés à l’annexe Section A.14 [Notes utilisées en percussion], page 688.

Notez bien que l’utilisation de hauteurs (comme un `cis4`) dans un contexte `DrumStaff` déclenchera inmanquablement une erreur. Un contexte `DrumStaff` fait automatiquement appel à une clef spécifique ; vous pouvez la spécifier explicitement ou utiliser une autre clef.

```
\drums {
  \clef percussion
  bd4 bd bd bd
  \clef treble
  hh4 hh hh hh
}
```



Certains instruments à percussion connaissent quelques problèmes quant à leur prise en charge au niveau de la génération de fichiers MIDI ; de plus amples informations sont disponibles à la rubrique Section 3.5.6 [MIDI et percussions], page 493.

Voir aussi

Manuel de notation : [Section 3.5.6 \[MIDI et percussions\]](#), page 493, [Section A.14 \[Notes utilisées en percussion\]](#), page 688.

Fichiers d'initialisation : `'ly/drumpitch-init.ly'`

Morceaux choisis : [Section "Percussion" dans *Morceaux choisis*](#).

Roulements de tambour

Les roulements de tambour s'indiquent par une triple barre en travers des hampes. Qu'il s'agisse d'une noire ou d'une durée plus longue, cette triple barre s'affiche explicitement. Dans le cas de croches, seules deux barres traversent les hampes (la troisième faisant office de ligature). Si ce roulement s'applique à des notes plus courtes que la croche, LilyPond fait apparaître une seule barre en travers des hampes en supplément du nombre de barres de ligature. Ces différents graphismes s'obtiennent à l'aide d'une notation de trémolo, en suivant les préceptes mentionnés à la rubrique [\[Répétitions en trémolo\]](#), page 147.

```
\drums {
  \time 2/4
  sn16 sn8 sn16 sn8 sn8:32 ~
  sn8 sn8 sn4:32 ~
  sn4 sn8 sn16 sn16
  sn4 r4
}
```



Les coups de baguette peuvent s'indiquer à l'aide de *markups* "D" et "G" au-dessus ou en dessous des notes comme indiqué à la rubrique [Section 5.4.2 \[Direction et positionnement\]](#), page 579. Vous devrez peut-être jouer sur la propriété `staff-padding` pour obtenir une ligne de base satisfaisante.

```
\drums {
  \repeat unfold 2 {
    sn16^"G" sn^"D" sn^"G" sn^"G" sn^"D" sn^"G" sn^"D" sn^"D"
    \stemUp
    sn16_"G" sn_"D" sn_"G" sn_"G" sn_"D" sn_"G" sn_"D" sn_"D"
  }
}
```



Voir aussi

Manuel de notation : [Section 5.4.2 \[Direction et positionnement\]](#), page 579, [\[Répétitions en trémolo\]](#), page 147.

Morceaux choisis : [Section "Percussion" dans *Morceaux choisis*](#).

Hauteurs en percussions

Certains instruments à percussion émettent des hauteurs, comme le xylophone, le vibraphone ou les timbales ; ils utilisent donc des portées classiques. Cette possibilité est abordée dans d'autres parties du manuel.

Voir aussi

Manuel de notation : [Section 3.5.6 \[MIDI et percussions\]](#), page 493.

Morceaux choisis : [Section “Percussion”](#) dans *Morceaux choisis*.

Portées de percussion

Une partie de percussions utilisant plusieurs instruments requiert en général une portée de plusieurs lignes, où chaque hauteur sur la portée représente un instrument à percussion.

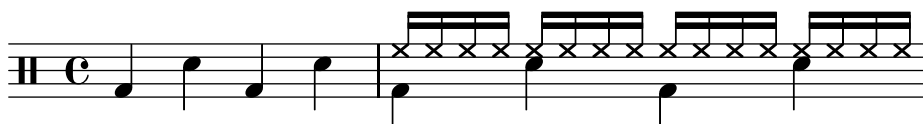
Pour saisir cette musique, il faut que les notes soient situées dans des contextes `DrumStaff` et `DrumVoice`.

```
up = \drummode {
  crashcymbal4 hihat8 halfopenhihat hh hh hh openhihat
}
down = \drummode {
  bassdrum4 snare8 bd r bd sn4
}
\new DrumStaff <<
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
>>
```



L'exemple ci-dessus montre une notation polyphonique détaillée. La notation polyphonique abrégée peut être employée – voir la rubrique [Section “J’entends des Voix”](#) dans *Manuel d’initiation* – comme ici :

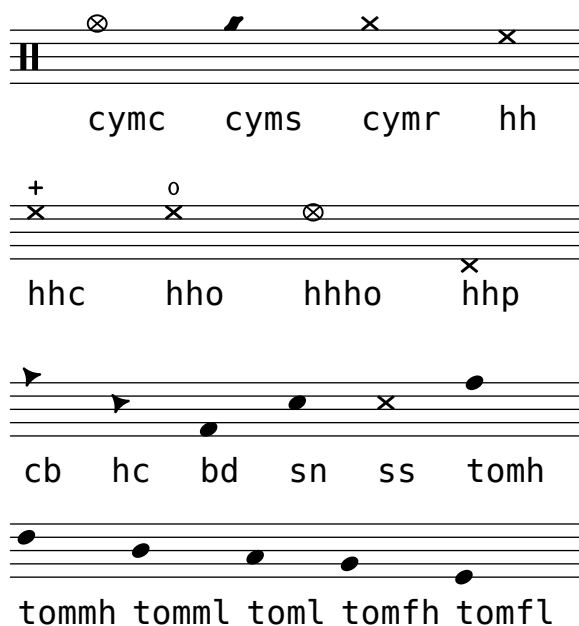
```
\new DrumStaff <<
  \drummode {
    bd4 sn4 bd4 sn4
    << {
      \repeat unfold 16 hh16
    } \\ {
      bd4 sn4 bd4 sn4
    } >>
  }
>>
```



On peut choisir d'autres mises en forme si l'on définit la propriété `drumStyleTable` dans le contexte `DrumVoice`. Quelques variables prédéfinies sont disponibles :

drums-style

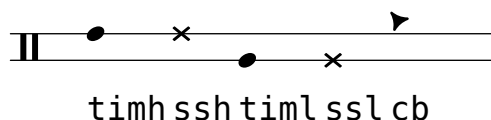
La notation par défaut : une batterie standard, sur une portée de cinq lignes.



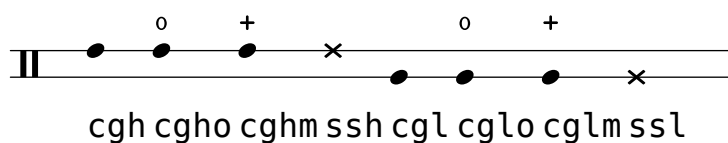
Le plan de la batterie peut inclure jusqu'à six toms différents. Bien sûr, vous n'êtes pas obligé de tous les utiliser si la musique en prévoit moins ; par exemple, les trois toms des lignes du milieu sont `tommh`, `tomml`, et `tomfh`.

timbales-style

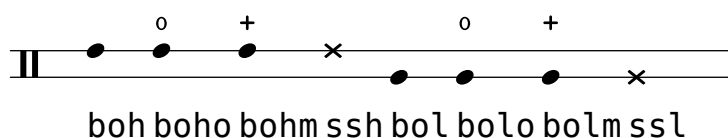
Ce style permet de saisir une partie de timbales, sur une portée à deux lignes.

**congas-style**

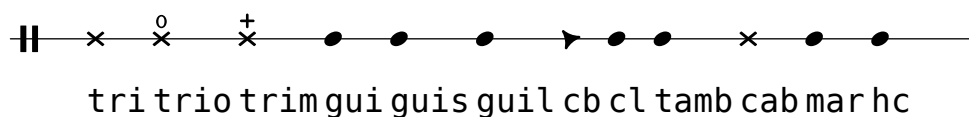
Ce style produit une portée à deux lignes pour une partie de congas.

**bongos-style**

Ce style produit une portée à deux lignes pour une partie de bongos.

**percussion-style**

Ce style permet de saisir toute sorte de percussions sur des portées d'une ligne.



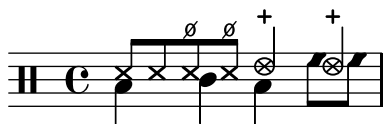
Personnalisation de portées de percussion

Dans l'éventualité où aucun des styles prédéfinis ne vous convient, il est aisé de définir le vôtre en début de fichier.

```
#(define mydrums '(
  (bassdrum      default  #f          -1)
  (snare         default  #f          0)
  (hihat         cross    #f          1)
  (halfopenhihat cross    "halfopen"  1)
  (pedalhihat    xcircle  "stopped"   2)
  (lowtom        diamond  #f          3)))

up = \drummode { hh8 hh hhho hhho hhp4 hhp }
down = \drummode { bd4 sn bd toml8 toml }

\new DrumStaff <<
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
>>
```



Morceaux choisis

Voici quelques exemples.

Deux *woodblocks* saisis comme *wbh* (woodblock aigu) et *wbl* (woodblock grave) :

```
% These lines define the position of the woodblocks in the stave;
% if you like, you can change it or you can use special note heads
% for the woodblocks.
#(define mydrums '((hiwoodblock default #t 3)
  (lowoodblock default #t -2)))

woodstaff = {
  % This defines a staff with only two lines.
  % It also defines the positions of the two lines.
  \override Staff.StaffSymbol #'line-positions = #'(-2 3)

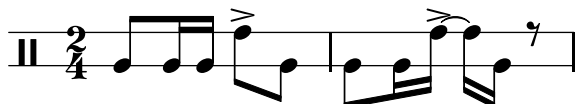
  % This is necessary; if not entered, the barline would be too short!
  \override Staff.BarLine #'bar-extent = #'(-1.5 . 1.5)
}

\new DrumStaff {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  % with this you load your new drum style table
  \woodstaff

  \drummode {
    \time 2/4
    wbl8 wbl16 wbl wbh8-> wbl |
```

```
wbl8 wbl16 wbh-> ~ wbh wbl16 r8 |
}
```



Vous aurez remarqué ici l'allongement des barres de mesures par l'instruction `\override Staff.BarLine #'bar-extent #'(de . à)`. Il faut aussi définir la position des deux lignes de la portée – voir à ce sujet la rubrique [\[Symbole de la portée\]](#), page 178.

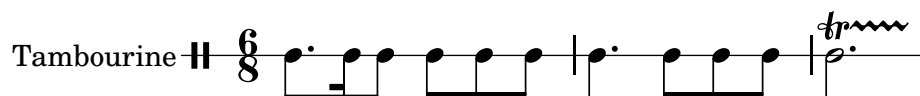
Un tambourin, saisi avec un « tamb » :

```
#(define mydrums '((tambourine default #t 0)))

tambustaff = {
  \override Staff.StaffSymbol #'line-positions = #'( 0 )
  \override Staff.BarLine #'bar-extent = #'(-1.5 . 1.5)
  \set DrumStaff.instrumentName = #"Tambourine"
}

\new DrumStaff {
  \tambustaff
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \drummode {
    \time 6/8
    tamb8. tamb16 tamb8 tamb tamb tamb |
    tamb4. tamb8 tamb tamb |
    % the trick with the scaled duration and the shorter rest
    % is necessary for the correct ending of the trill-span!
    tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
  }
}
```



Un peu de tam tam, abrégé « tt » :

```
#(define mydrums '((tamtam default #t 0)))

tamtamstaff = {
  \override Staff.StaffSymbol #'line-positions = #'( 0 )
  \override Staff.BarLine #'bar-extent = #'(-1.5 . 1.5)
  \set DrumStaff.instrumentName = #"Tamtam"
}


\new DrumStaff {
  \tamtamstaff
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \drummode {
```

```

    tt 1 \pp \laissezVibrer
  }
}

```

Tamtam 

Deux cloches, une sonnaillle – « cb » pour *cowbell* – et une cloche de quart – « rb » pour *ridebell* :

```

#(define mydrums '((ridebell default #t 3)
                   (cowbell default #t -2)))

bellstaff = {
  \override DrumStaff.StaffSymbol #'line-positions = #'(-2 3)
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.BarLine #'bar-extent = #'(-1.5 . 1.5)
  \set DrumStaff.instrumentName = #"Different Bells"
}

\new DrumStaff {
  \bellstaff
  \drummode {
    \time 2/4
    rb8 rb cb cb16 rb-> ~ |
    rb16 rb8 rb16 cb8 cb |
  }
}

```

Different Bells 

Et pour finir un extrait tiré de « L’histoire du Soldat » de Stravinsky :

```

#(define mydrums '((bassdrum default #t 4)
                   (snare default #t -4)
                   (tambourine default #t 0)))

global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
  \time 2/4 s2
}

drumsA = {
  \context DrumVoice <<
    { \global }
    { \drummode {
      \autoBeamOff
      \stemDown sn8 \stemUp tamb s8 |
      sn4 \stemDown sn4 |
      \stemUp tamb8 \stemDown sn8 \stemUp sn16 \stemDown sn \stemUp sn8 |
    }
  }
}

```

```

        \stemDown sn8 \stemUp tamb s8 |
        \stemUp sn4 s8 \stemUp tamb
    }
}
>>
}

drumsB = {
  \drummode {
    s4 bd8 s2*2 s4 bd8 s4 bd8 s8
  }
}

\layout {
  indent = #40
}

\score {
  \new StaffGroup <<
    \new DrumStaff {
      \set DrumStaff.instrumentName = \markup {
        \column {
          "Tambourine"
          "et"
          "caisse claire s. timbre"
        }
      }
      \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
      \drumsA
    }

    \new DrumStaff {
      \set DrumStaff.instrumentName = #"Grosse Caisse"
      \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
      \drumsB }
  >>
}

```

Tambourine
et
caisse claire s. timbre

Grosse Caisse



Voir aussi

Morceaux choisis : [Section “Percussion”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “DrumStaff”](#) dans *Référence des propriétés internes*, [Section “DrumVoice”](#) dans *Référence des propriétés internes*.

Notes fantômes

Des notes fantômes peuvent être créées pour les parties de percussion, grâce à la commande `\parenthesize` décrite à la rubrique [\[Parenthèses\]](#), page 206. Cependant, le mode `\drummode` n'inclut pas par défaut le graveur `Parenthesis_engraver` qui permet d'imprimer ces signes.

```
\new DrumStaff \with {
  \consists "Parenthesis_engraver"
}
<<
\context DrumVoice = "1" { s1 }
\context DrumVoice = "2" { s1 }
\drummode {
  <<
  {
    hh8[ hh] <hh sn> hh16
    < \parenthesize sn > hh
    < \parenthesize sn > hh8 <hh sn> hh
  } \\\
  {
    bd4 r4 bd8 bd r8 bd
  }
  >>
}
>>
```



Notez que les commandes `\parenthesize` obligent à ajouter des accords – sous la forme `< >` – autour de chaque élément.

Voir aussi

Morceaux choisis : [Section “Percussion”](#) dans *Morceaux choisis*.

2.6 Instruments à vent

Moderato assai

Ce chapitre traite de certains aspects particuliers en matière de notation pour instruments à vent.

2.6.1 Vue d'ensemble des instruments à vent

Nous allons aborder ici quelques aspects communs à la plupart des instruments à vent.

Références en matière d'instruments à vent

Ce qui caractérise les partitions pour instruments à vent a trait principalement à la respiration et à l'attaque :

- Les respirations s'indiquent par des silences ou des [Signes de respiration], page 123.
- Un jeu *legato* s'indique par des [Liaisons d'articulation], page 119.
- Les différents types d'attaque – *legato*, détaché ou piqué – s'indiquent en principe par des signes d'articulation, parfois agrémentés de liaisons. Voir à ce sujet [Articulations et ornements], page 109 et Section A.13 [Liste des signes d'articulation], page 687.
- Un *Flutterzunge* (trémolo dental ou trémolo en roulant les r) s'indique par une marque de trémolo et une étiquette textuelle attachée à la note concernée. Voir à ce sujet [Répétitions en trémolo], page 147.

D'autres aspects de la notation s'appliquent aussi aux instruments à vent :

- De nombreux instruments à vent sont transpositeurs ; voir [Instruments transpositeurs], page 22.
- Les glissandos sont l'une des caractéristiques du trombone à coulisse, bien que d'autres instruments puissent y parvenir en jouant sur les pistons ou des clés ; consulter [Glissando], page 125.
- Des glissandos harmoniques sont réalisables par les cuivres. Ils sont traditionnellement indiqués par des [Notes d'ornement], page 102.
- Les inflexions en fin de note sont abordées au chapitre [Chutes et sauts], page 124.
- Les « bruitages » de clé ou de piston s'indiquent souvent par le style **cross** ou des [Têtes de note spécifiques], page 33.
- Les bois peuvent émettre des harmoniques dans le bas de leur registre. On les indique avec un **flageolet** – voir Section A.13 [Liste des signes d'articulation], page 687.
- En ce qui concerne les cuivres, la sourdine s'indique en principe par une étiquette textuelle. Cependant, lorsque les changements sont nombreux et rapides, il est d'usage de recourir aux articulations **stopped** et **open**. Pour de plus amples détails, voir [Articulations et ornements], page 109 et Section A.13 [Liste des signes d'articulation], page 687.
- La sourdine du cor d'harmonie s'indique par un **stopped**. Voir le chapitre [Articulations et ornements], page 109.

Morceaux choisis

Modifier la taille d'un \flageolet

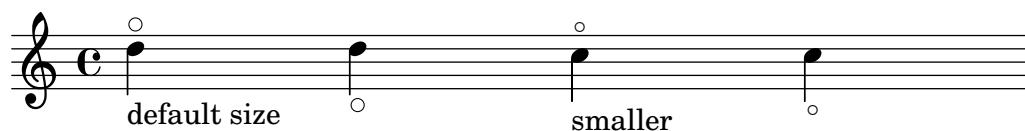
Il est possible de rapetisser le cercle d'un `\flageolet` grâce à une fonction Scheme.

```
smallFlageolet =
#(let ((m (make-articulation "flageolet")))
  (set! (ly:music-property m 'tweaks)
    (acons 'font-size -3
      (ly:music-property m 'tweaks)))
  m)

\layout { ragged-right = ##f }

\relative c' {
```

```
d4^\flageolet_\markup { default size } d_\flageolet
c4^\smallFlageolet_\markup { smaller } c_\smallFlageolet
}
```



Voir aussi

Manuel de notation : [Signes de respiration], page 123, [Liaisons d'articulation], page 119, [Articulations et ornements], page 109, Section A.13 [Liste des signes d'articulation], page 687, [Répétitions en trémolo], page 147, [Instruments transpositeurs], page 22, [Glissando], page 125, [Notes d'ornement], page 102, [Chutes et sauts], page 124, [Têtes de note spécifiques], page 33.

Morceaux choisis : Section “Vents” dans *Morceaux choisis*.

Doigtés pour vents

Tout instrument à vent, hormis le trombone à coulisse, fait appel à plusieurs doigts pour produire un son. Les exemples ci-dessous vous donnent un aperçu de différentes manières d'indiquer des doigtés.

La gestion des diagrammes de doigté spécifiques aux bois est abordée plus en profondeur au chapitre Section 2.6.3.1 [Diagrammes pour bois], page 377.

Morceaux choisis

Symboles de doigtés pour instruments à vent

Des symboles spécifiques peuvent être obtenus en combinant les glyphs disponibles, ce qui est tout à fait indiqué en matière d'instrument à vent.

```
centermarkup = {
  \once \override TextScript #'self-alignment-X = #CENTER
  \once \override TextScript #'X-offset =#(ly:make-simple-closure
    `(+
      ,(ly:make-simple-closure (list
        ly:self-alignment-interface::centered-on-x-parent))
      ,(ly:make-simple-closure (list
        ly:self-alignment-interface::x-aligned-on-self))))
}
\score
{\relative c'
  {
    g\open
    \once \override TextScript #'staff-padding = #-1.0 \centermarkup
    g^\markup{\combine \musicglyph #"scripts.open" \musicglyph
      #"scripts.tenuto"}
    \centermarkup g^\markup{\combine \musicglyph #"scripts.open"
      \musicglyph #"scripts.stopped"}
    g\stopped
  }
}
```

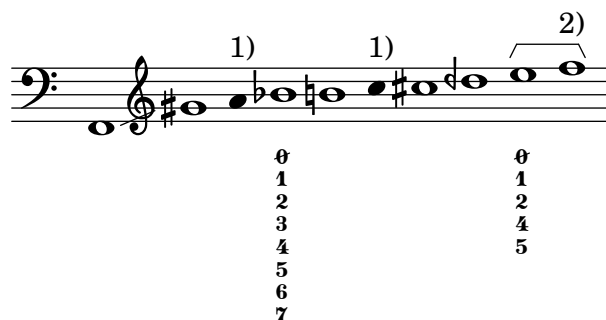
*Doigtés pour flûte à bec*

Cet exemple illustre la manière de créer et afficher des indications de doigté pour instrument à vent.

```
% range chart for paetzold contrabass recorder
```

```
centermarkup = {
  \once \override TextScript #'self-alignment-X = #CENTER
  \once \override TextScript #'X-offset =#(ly:make-simple-closure
    `(+
      , (ly:make-simple-closure (list
        ly:self-alignment-interface::centered-on-x-parent))
      , (ly:make-simple-closure (list
        ly:self-alignment-interface::x-aligned-on-self))))
}

\score {
  \new Staff \with {
    \remove "Time_signature_engraver"
    \override Stem #'stencil = ##f
    \override Flag #'stencil = ##f
    \consists "Horizontal_bracket_engraver"
  }
  {
    \clef bass
    \set Score.timing = ##f
    f,1*1/4 \glissando
    \clef violin
    gis'1*1/4
    \stemDown a'4^\markup{1)}
    \centermarkup
    \once \override TextScript #'padding = #2
    bes'1*1/4_\markup{\override #'(baseline-skip . 1.7) \column
      { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2 \finger 3 \finger 4
        \finger 5 \finger 6 \finger 7} }
    b'1*1/4
    c''4^\markup{1)}
    \centermarkup
    \once \override TextScript #'padding = #2
    cis''1*1/4
    deh''1*1/4
    \centermarkup
    \once \override TextScript #'padding = #2
    \once \override Staff.HorizontalBracket #'direction = #UP
    e''1*1/4_\markup{\override #'(baseline-skip . 1.7) \column
      { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2 \finger 4
        \finger 5} }\startGroup
    f''1*1/4^\markup{2)}\stopGroup
  }
}
```

Voir aussi

Manuel de notation : [Section 2.6.3.1 \[Diagrammes pour bois\]](#), page 377.

Morceaux choisis : [Section “Vents”](#) dans *Morceaux choisis*.

2.6.2 Cornemuse

Voici quelques informations spécifiques à la cornemuse.

Définitions pour la cornemuse

LilyPond inclut des définitions spécifiques destinées à la notation pour cornemuse écossaise ; pour les utiliser, il suffit d'ajouter

```
\include "bagpipe.ly"
```

en début de fichier. Vous bénéficierez ainsi de commandes courtes pour les appoggiatures spéciales et idiomatiques de la cornemuse. Par exemple, `\taor` est un raccourci pour

```
\grace { \small G32[ d G e ] }
```

‘`bagpipe.ly`’ prend également en charge les définitions de hauteurs pour la cornemuse ; vous n’avez donc pas à vous soucier d’employer `\relative` ou `\transpose`.

```
\include "bagpipe.ly"
```

```
{ \grg G4 \grg a \grg b \grg c \grg d \grg e \grg f \grA g A }
```



La musique pour cornemuse est traditionnellement écrite en ré majeur. Dans la mesure où c’est la seule tonalité possible, l’usage est de ne pas indiquer l’armure explicitement. À cet effet, pensez à commencer votre partition par `\hideKeySignature` – ou bien `\showKeySignature` si, pour une raison ou pour une autre, vous tenez à afficher l’armure.

Des compositions actuelles peuvent ne pas respecter cette tonalité traditionnelle, auquel cas les `do` et `fa` devraient être abaissés en utilisant `cflat` ou `fflat` ; ils seront représentés par une note en forme de croix.

Lorsqu’on joue des œuvres moins cérémonieuses que pour une parade ou un défilé, peut survenir un `sol` aigu, au doux nom de « *piobaireachd* », et que l’on indiquera par `gflat`.

Voir aussi

Morceaux choisis : [Section “Vents”](#) dans *Morceaux choisis*.

Exemple pour la cornemuse

Et voici en guise d’exemple, à quoi ressemble le chant populaire *Amazing Grace*, noté dans l’idiome de la cornemuse.

```

\include "bagpipe.ly"
\layout {
  indent = 0.0\cm
  \context { \Score \remove "Bar_number_engraver" }
}

\header {
  title = "Amazing Grace"
  meter = "Hymn"
  arranger = "Trad. arr."
}

{
  \hideKeySignature
  \time 3/4
  \grg \partial 4 a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg e8. f16
  \dblA A2 \grg A4
  \grg A2 f8. A16
  \grg A2 \hdbl f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 e4
  \thrwd d2.
  \slurd d2
  \bar "|."
}

```

Amazing Grace

Hymn

Trad. arr.




```

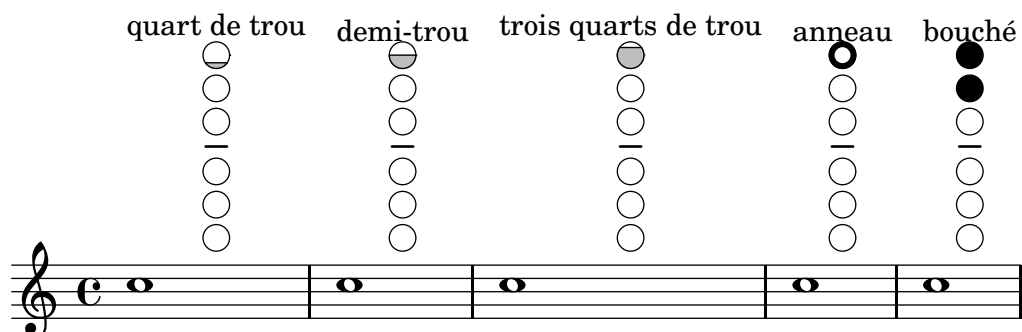
\center-column {
  "demi-trou"
  \woodwind-diagram #'flute #'((cc . (one1h))
                        (lh . ()))
                        (rh . ()))
}
}

c1^\markup {
  \center-column {
    "trois quarts de trou"
    \woodwind-diagram #'flute #'((cc . (one3q))
                                    (lh . ()))
                                    (rh . ()))
  }
}

c1^\markup {
  \center-column {
    "anneau"
    \woodwind-diagram #'flute #'((cc . (oneR))
                                    (lh . ()))
                                    (rh . ()))
  }
}

c1^\markup {
  \center-column {
    "bouché"
    \woodwind-diagram #'flute #'((cc . (oneF two))
                                    (lh . ()))
                                    (rh . ()))
  }
}

```



L'indication du doigté permettant de triller s'obtient en grisant une position :

```

c1^\markup {
  \woodwind-diagram #'bass-clarinet
    #'((cc . (threeT four))
        (lh . ()))
        (rh . (b fis)))
}

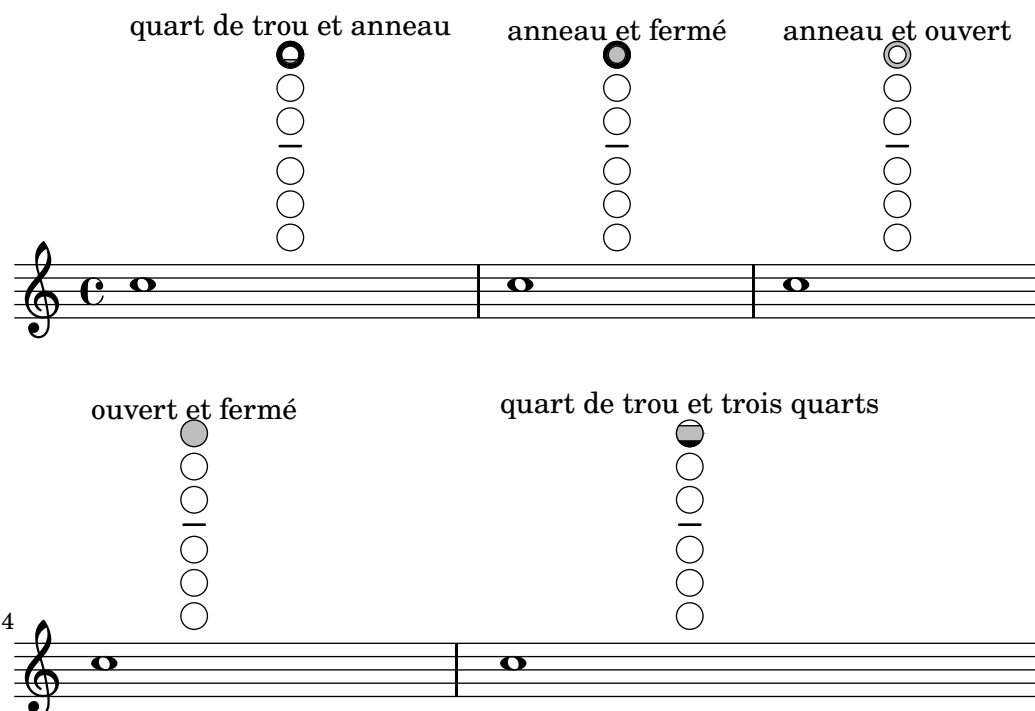
```



```

\center-column {
  "quart de trou et trois quarts"
  \woodwind-diagram #'flute #'((cc . (one1qT3q))
                        (lh . ()))
                        (rh . ()))
}

```



Bien que cela ne produise pas de notation, vous pouvez obtenir la liste de toutes les possibilités pour un instrument donné, en utilisant dans un fichier les instructions `$(print-keys-verbose 'vent)` – affichage à l'écran – ou `$(print-keys-verbose 'vent (current-error-port))` – génération d'un fichier de journalisation.

De nouveaux diagrammes sont réalisables, bien que ceci requiert de maîtriser le langage Scheme et n'est pas à la portée de tous les utilisateurs. Des gabarits sont contenus dans les fichiers `'scm/define-woodwind-diagrams.scm'` et `'scm/display-woodwind-diagrams.scm'`.

Morceaux choisis

Liste des diagrammes de doigtés pour bois

Voici les différents instruments à vent de la section des bois pour lesquels LilyPond peut, à ce jour, afficher des doigtés.

```

\relative c' {
  \textLengthOn
  c1~
  \markup {
    \center-column {
      'piccolo
      " "
      \woodwind-diagram
        #'piccolo
        #'()
    }
  }
}

```

```

    }
}

c1~
\markup {
  \center-column {
    'flute
    " "

    \woodwind-diagram
    #'flute
    #'()
  }
}

c1~\markup {
  \center-column {
    'oboe
    " "

    \woodwind-diagram
    #'oboe
    #'()
  }
}

c1~\markup {
  \center-column {
    'clarinet
    " "

    \woodwind-diagram
    #'clarinet
    #'()
  }
}

c1~\markup {
  \center-column {
    'bass-clarinet
    " "

    \woodwind-diagram
    #'bass-clarinet
    #'()
  }
}

c1~\markup {
  \center-column {
    'saxophone
    " "

    \woodwind-diagram
    #'saxophone
    #'()
  }
}

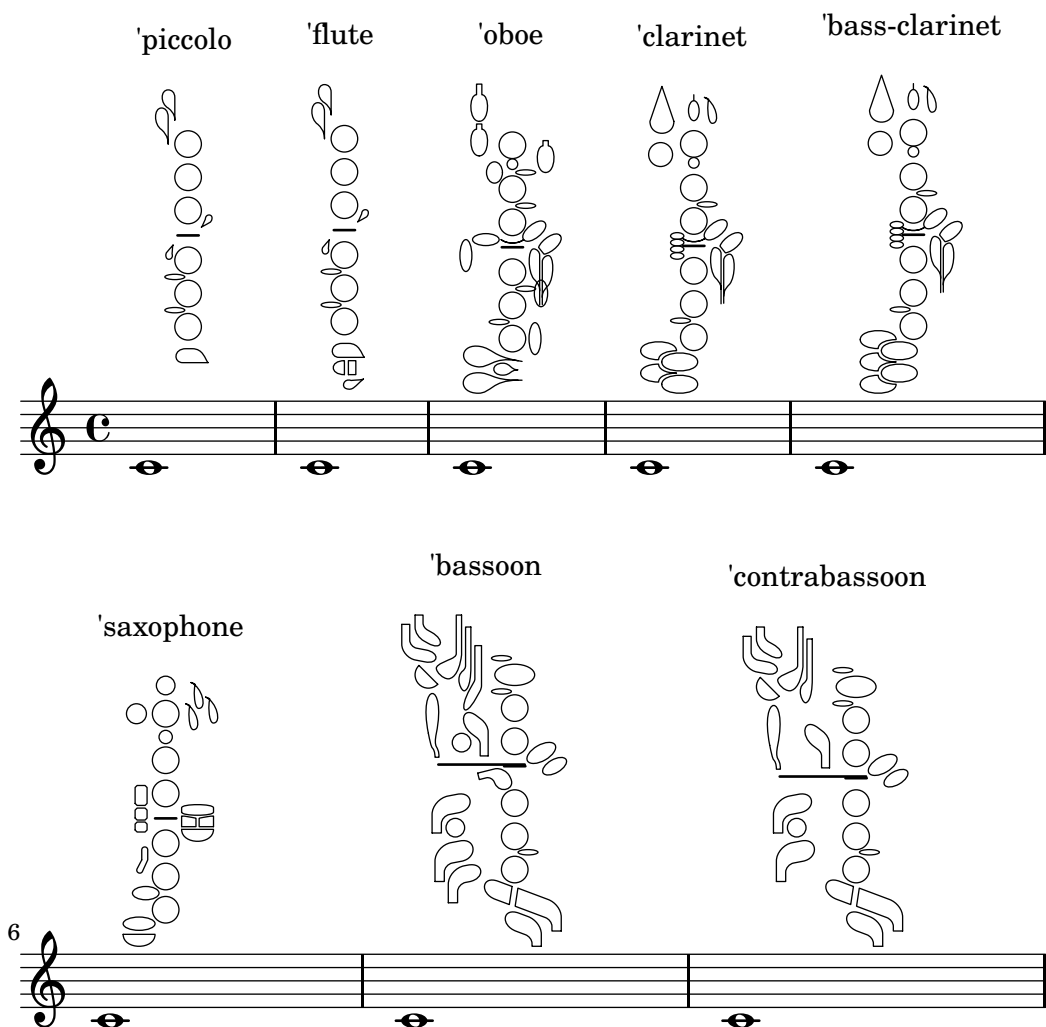
```

```

c1^\markup {
  \center-column {
    'bassoon
    " "
    \woodwind-diagram
    #'bassoon
    #'()
  }
}

c1^\markup {
  \center-column {
    'contrabassoon
    " "
    \woodwind-diagram
    #'contrabassoon
    #'()
  }
}
}

```

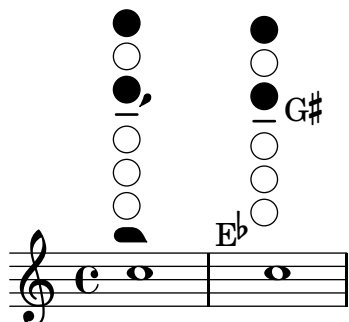


Ajout de texte à un diagramme de doigté

Dans certains cas, vous pouvez opter pour l’affichage textuel d’une clé située à côté d’un trou plutôt que sa représentation graphique.

```
\relative c'' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram
      #'piccolo
      #'((cc . (one three))
        (lh . (gis))
        (rh . (ees)))

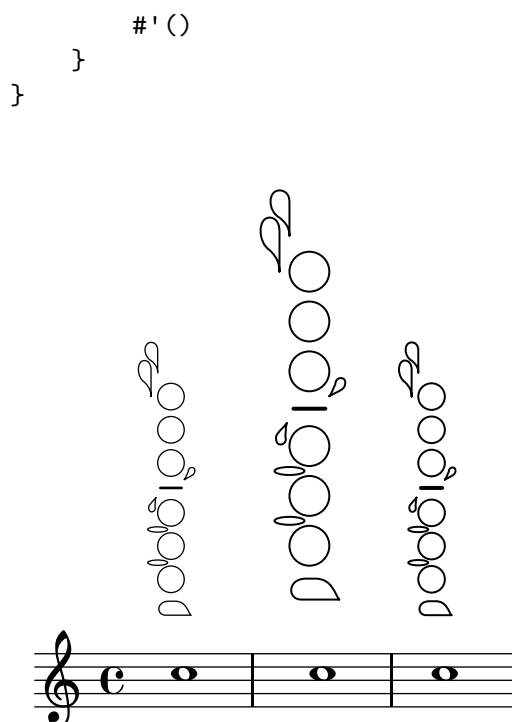
  c^\markup
    \override #'(graphical . #f) {
      \woodwind-diagram
        #'piccolo
        #'((cc . (one three))
          (lh . (gis))
          (rh . (ees)))
    }
}
```

*Modification de la taille d'un diagramme pour bois*

La taille et l'épaisseur des diagrammes de doigté pour bois est modifiable à souhait.

```
\relative c'' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram
      #'piccolo
      #'()

  c^\markup
    \override #'(size . 1.5) {
      \woodwind-diagram
        #'piccolo
        #'()
    }
  c^\markup
    \override #'(thickness . 0.15) {
      \woodwind-diagram
        #'piccolo
    }
}
```



Liste des différents diagrammes de doigtés pour bois

Le code suivant permet d'obtenir une liste de toutes les possibilités en matière de doigtés pour bois, tels qu'ils sont définis dans le fichier 'scm/define-woodwind-diagrams.scm'. Cette liste sera produite en dans le fichier de journalisation, mais pas sous forme de musique. Pour un affichage en console, supprimez la partie (current-error-port) des commandes.

```
#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboe (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
#(print-keys-verbose 'alto-saxophone (current-error-port))
#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))
```

Voir aussi

Fichiers d'initialisation : 'scm/define-woodwind-diagrams.scm',
'scm/display-woodwind-diagrams.scm'.

Morceaux choisis : [Section "Vents"](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section "TextScript"](#) dans *Référence des propriétés internes*, [Section "instrument-specific-markup-interface"](#) dans *Référence des propriétés internes*.

2.7 Notation des accords

The image shows two systems of musical notation. The first system has two staves (treble and bass clef) with lyrics: "1. Fair is the sun - shine, Fair - er the moon - light" and "2. Fair are the mead - ows, Fair - er the wood - land,". Above the first staff are chord symbols: F, C, F, F, C, F. The second system also has two staves with lyrics: "And all the stars in heav'n a - bove;" and "Robed in the flow - ers of bloom - ing spring;". Above the first staff are chord symbols: F, Bb, F, C7, F, C.

Les accords se saisissent soit comme des notes normales, soit à l'aide d'un mode dédié, et seront rendus selon l'une des diverses nomenclatures occidentales. Les accords peuvent aussi se présenter sous forme nominale, ou bien en basse figurée.

2.7.1 Mode accords

Le mode accords permet de saisir des accords en indiquant leur structure plutôt que les notes qui les composent.

Généralités sur le mode accords

Un accord peut se saisir en tant que musique simultanée, comme nous l'avons vu à la rubrique [\[Notes en accords\]](#), page 149.

LilyPond dispose aussi d'un « mode accords » au sein duquel sera considérée la structure des accords, tels qu'ils se présentent dans les traditions occidentales, plutôt que les différentes hauteurs qui les composent. Ce mode est tout à fait adapté pour ceux qui sont plus habitués aux accords nommés. Pour plus d'information quant aux différentes façons de libeller votre code, voir [Section 5.4.1 \[Modes de saisie\]](#), page 577.

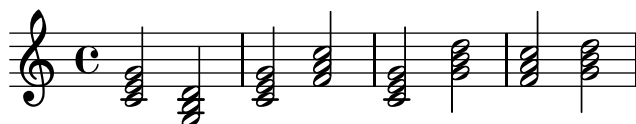
```
\chordmode { c1 g a g c }
```

The image shows a single staff in treble clef with a common time signature 'C'. It contains five chords represented by block letters: C, G, A, G, and C. Each chord is preceded by a sharp sign (#) and a block letter, indicating absolute pitch.

Tout accord saisi dans ce mode dédié constitue un élément musical à part entière ; il pourra donc par exemple être transposé comme n'importe quel ensemble de hauteurs simultanées. `\chordmode` travaille en absolu ; une instruction `\relative` restera sans effet au sein d'un bloc `chordmode`. Notez toutefois que les hauteurs absolues sont une octave plus haut en `\chordmode` qu'en mode notes traditionnel.

Mode notes et mode accords peuvent tout à fait cohabiter dans une séquence musicale :

```
<c e g>2 <g b d>
\chordmode { c2 f }
<c e g>2 <g' b d>
\chordmode { f2 g }
```



Voir aussi

Glossaire musicologique : [Section “Accord”](#) dans *Glossaire*.

Manuel de notation : [\[Notes en accords\]](#), page 149, [Section 5.4.1 \[Modes de saisie\]](#), page 577.

Morceaux choisis : [Section “Accords”](#) dans *Morceaux choisis*.

Problèmes connus et avertissements

Les raccourcis d’articulation ou d’ornementation ne sont pas disponibles en mode accord – voir [\[Articulations et ornements\]](#), page 109.

Lorsque vous mélangez mode accords et mode notes dans une séquence musicale qui démarre par le mode accords, le mode notes créera un nouveau contexte **Staff** :

```
\chordmode { c2 f }
<c e g>2 <g' b d>
```



La solution consiste à créer explicitement le contexte **Staff** dès le départ :

```
\new Staff {
  \chordmode { c2 f }
  <c e g>2 <g' b d>
}
```



Accords courants

Dans le mode accords, introduit par la commande `\chordmode`, les accords ne sont indiqués que par leur note fondamentale, à laquelle on peut adjoindre une durée.

```
\chordmode { c2 f4 g }
```



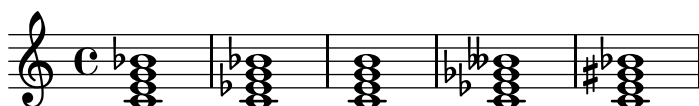
On peut cependant obtenir des accords différents – mineur, augmenté ou diminué – en utilisant le caractère deux points (:).

```
\chordmode { c2:m f4:aug g:dim }
```



Les accords de septième sont aisément stipulables :

```
\chordmode { c1:7 c:m7 c:maj7 c:dim7 c:aug7 }
```



Voici les différents modificateurs d'accord de trois sons ou de septième. Par défaut, la septième ajoutée sera mineure ; la septième de dominante est donc l'accord de septième de base. Toutes les altérations sont relatives à la septième de dominante. Une table étendue des modificateurs et de leur utilisation est à l'annexe [Section A.2 \[Modificateurs d'accord usuels\]](#), page 606.

Modificateur	Action	Exemple
Néant	Action par défaut ; produit une triade majeure.	
m, m7	Accord mineur. Ce modificateur abaisse la tierce, et la septième s'il y en a une.	
dim, dim7	Accord diminué. Ce modificateur minorise la tierce, diminue la quinte et la septième s'il y en a.	
aug	Accord augmenté. Ce modificateur augmente la quinte.	
maj, maj7	Accord de septième majeure. Ce modificateur majorise la septième. Le 7 à la suite du est maj facultatif. Ce modificateur ne sert pas à créer une triade majeure.	

Voir aussi

Manuel de notation : [\[Extension et altération d'accords\]](#), page 388, [Section A.2 \[Modificateurs d'accord usuels\]](#), page 606.

Morceaux choisis : [Section "Accords" dans Morceaux choisis](#).

Problèmes connus et avertissements

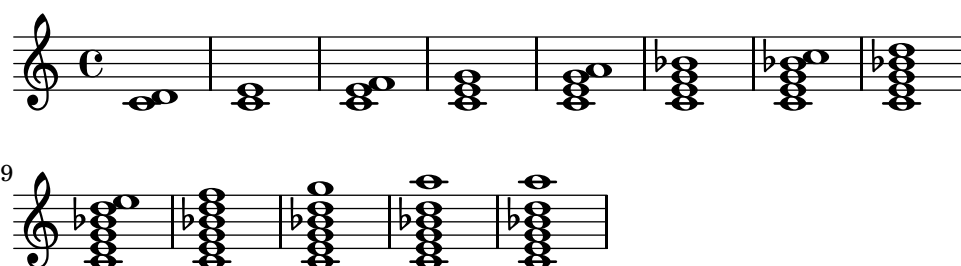
Un accord ne devrait comporter qu'un seul modificateur de qualité. La présence de plusieurs modificateurs ne déclenchera ni avertissement, ni erreur, mais le résultat pourrait être surprenant. Un accord qui n'est pas constructible à l'aide d'un unique modificateur devra faire l'objet d'une altération de ses composantes, comme indiqué à la rubrique [\[Extension et altération d'accords\]](#), page 388.

Extension et altération d'accords

Le mode accords permet d'élaborer des accords complexes. Ainsi, on peut enrichir l'accord par des notes ajoutées, enlever certaines notes, augmenter ou diminuer certains intervalles, ajouter la note de basse ou créer un renversement.

Le premier nombre qui suit le caractère deux-points (:) permet de déterminer l'étendue d'un accord. L'accord sera construit par ajout à la fondamentale, d'autant de tierces consécutives que nécessaire pour atteindre le nombre spécifié. N'oubliez pas que la septième ajoutée à un accord est minorée par défaut. Lorsque le dernier degré ne correspond pas à une tierce – la sixte par exemple – les tierces seront empilées jusqu'à celle directement inférieure au degré mentionné, qui sera alors ajouté pour conclure l'accord. L'étendue maximale autorisée est la treizième ; toute étendue plus élevée sera interprétée comme un accord de treizième.

```
\chordmode {
  c1:2 c:3 c:4 c:5
  c1:6 c:7 c:8 c:9
  c1:10 c:11 c:12 c:13
  c1:14
}
```



Notez bien que c:5 et c produisent tous deux une triade de do majeur.

Dans la mesure où un accord de treizième majeure ne sonne pas très bien avec la onzième, la onzième est automatiquement enlevée, sauf à l'avoir explicitement spécifiée.

```
\chordmode {
  c1:13 c:13.11 c:m13
}
```



On peut enrichir l'accord par des notes ajoutées, que l'on indique après le chiffrage principal et que l'on sépare par des points, sans oublier que si l'on y ajoute une septième, celle-ci sera minorée et non majeure.

```
\chordmode {
  c1:5.6 c:3.7.8 c:3.6.13
}
```



Les notes ajoutées peuvent monter aussi haut que de besoin.

```
\chordmode {
  c4:5.15 c:5.20 c:5.25 c:5.30
}
```



On peut augmenter ou diminuer certains intervalles au moyen des signes - ou + au degré considéré. L'altération de l'un des degrés automatiquement inclus dans la structure de base d'un accord s'effectue de la même manière.

```
\chordmode {
  c1:7+ c:5+.3- c:3-.5-.7-
}
```



Après avoir ajouté des notes à un accord, on peut aussi en enlever certaines, en les spécifiant derrière un signe ^ – les séparer par un point lorsqu'il y en a plus d'une.

```
\chordmode {
  c1^3 c:7^5 c:9^3 c:9^3.5 c:13.11^3.7
}
```



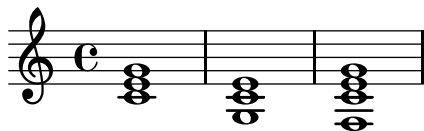
L'ajout du modificateur **sus** permet de créer un accord de suspension. Ceci aura pour effet de supprimer la tierce et y ajouter, suivant que vous spécifiez 2 ou 4, la seconde ou la quarte. **sus** est équivalent à ^3 ; **sus4** est équivalent à .4^3.

```
\chordmode {
  c1:sus c:sus2 c:sus4 c:5.4^3
}
```



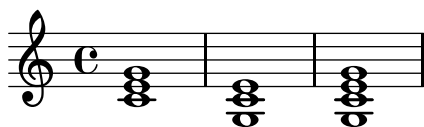
Les accords peuvent être renversés ou combinés avec une note étrangère à la basse, au moyen de **accord/hauteur**.

```
\chordmode {
  c1 c/g c/f
}
```



Si la note de basse précisée appartient à l'accord, la doublure supérieure sera supprimée. Pour l'éviter, utilisez la syntaxe */+hauteur*.

```
\chordmode {
  c1 c/g c/+g
}
```



Une table étendue des modificateurs et de leur utilisation est à l'annexe [Section A.2 \[Modificateurs d'accord usuels\]](#), page 606.

Voir aussi

Manuel de notation : [Section A.2 \[Modificateurs d'accord usuels\]](#), page 606.

Morceaux choisis : [Section "Accords" dans Morceaux choisis](#).

Problèmes connus et avertissements

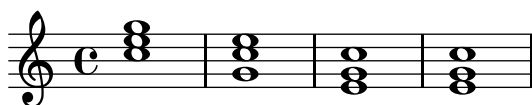
Aucun nom de note ne peut être indiqué deux fois dans un accord. Ainsi, dans l'accord suivant, seule la quinte augmentée est prise en compte, parce qu'elle est indiquée en dernier :

```
\chordmode { c1:5.5-.5+ }
```



Seul le deuxième renversement peut être obtenu par adjonction de la basse. Le premier renversement requiert de changer la fondamentale de l'accord.

```
\chordmode {
  c'1: c':/g e:6-3-^5 e:m6-^5
}
```



2.7.2 Gravure des accords

Les accords peuvent se présenter aussi bien sous forme nominative que comme un empilement de notes sur une portée.

Impression des noms d'accord

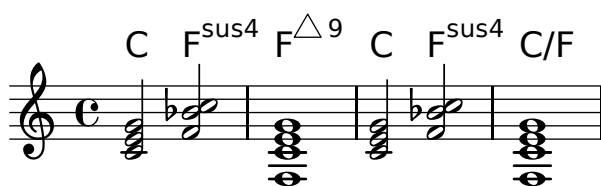
Les chiffrages d'accords sont liés au contexte `ChordNames` :

```
\new ChordNames {
  \chordmode {
    c2 f4. g8
  }
}
```

C F G

Les accords peuvent être saisis soit en tant que hauteurs simultanées, soit au moyen du mode accords. Le chiffrage affiché sera identique quel que soit le mode de saisie, à moins qu'il n'y ait inversion ou ajout de la basse.

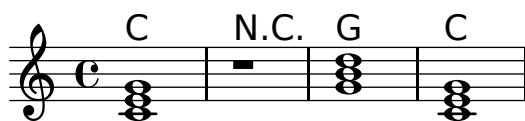
```
<<
\new ChordNames {
  <c e g>2 <f bes c>
  <f c' e g>1
  \chordmode {
    c2 f:sus4 c1:/f
  }
}
{
  <c e g>2 <f bes c>
  <f, c' e g>1
  \chordmode {
    c2 f:sus4 c1:/f
  }
}
>>
```



L'apparition de silences dans un contexte `ChordNames` déclenchera l'impression d'un *markup* `noChordSymbol`.

```
<<
\new ChordNames \chordmode {
  c1
  r1
  g1
  c1
}
\chordmode {
  c1
  r1
  g1
  c1
}
```

>>



`\chords { ... }` est un raccourci de `\new ChordNames { \chordmode { ... } }`.

```
\chords {
  c2 f4.:m g8:maj7
}
```

C Fm G[△]

```
\new ChordNames {
  \chordmode {
    c2 f4.:m g8:maj7
  }
}
```

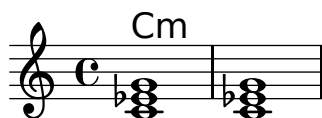
C Fm G[△]

Morceaux choisis

Impression des accords si changement

Vous pouvez faire ressortir les chiffres d'accords s'ils ne sont imprimés qu'aux changements d'accord ou en début de ligne.

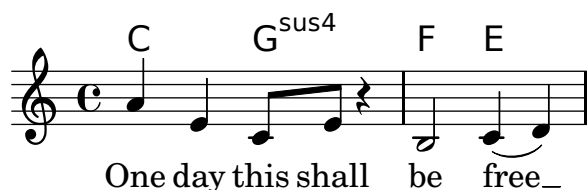
```
harmonies = \chordmode {
  c1:m c:m \break c:m c:m d
}
<<
\new ChordNames {
  \set chordChanges = ##t
  \harmonies
}
\new Staff {
  \relative c' { \harmonies }
}
>>
```



Chanson simple

Assembler des noms d'accords, une mélodie et des paroles permet d'obtenir la partition d'une chanson :

```
<<
\chords { c2 g:sus4 f e }
\relative c'' {
  a4 e c8 e r4
  b2 c4( d)
}
\addlyrics { One day this shall be free __ }
>>
```



Voir aussi

Glossaire musicologique : [Section “Accord” dans *Glossaire*](#).

Manuel de notation : [\[Saisie de musique en parallèle\]](#), page 168.

Morceaux choisis : [Section “Accords” dans *Morceaux choisis*](#).

Référence des propriétés internes : [Section “ChordNames” dans *Référence des propriétés internes*](#), [Section “ChordName” dans *Référence des propriétés internes*](#), [Section “Chord_name-engraver” dans *Référence des propriétés internes*](#), [Section “Volta-engraver” dans *Référence des propriétés internes*](#), [Section “Bar-engraver” dans *Référence des propriétés internes*](#).

Problèmes connus et avertissements

Un accord contenant un renversement ou dont la basse est changée ne sera pas chiffré correctement s’il est saisi en tant que musique simultanée.

Personnalisation des noms d’accord

Il existe plus d’un système de chiffrage d’accords. Le nom des accords varie selon les traditions musicales, et plusieurs symboles représentent un même accord. LilyPond vous permet de créer votre propre nomenclature d’accords, tant au niveau des noms que des symboles qui les représenteront.

Le modèle par défaut des chiffrages d’accord est celui de Klaus Ignatzek pour le jazz (cf. [Section “Références bibliographiques” dans *Essai*](#)). Il est possible de créer votre propre modèle de chiffrages à l’aide des différentes propriétés mentionnées ci-dessous. LilyPond dispose d’un système alternatif de chiffrage jazz qui a été élaboré grâce à ces mêmes propriétés. Les deux notations, Ignatzek et alternative, sont illustrés à l’annexe [Section A.1 \[Table des noms d’accord\]](#), page 605.

En plus des différents systèmes de nommage, le nom de la fondamentale varie selon la langue utilisée. Les instructions `\germanChords`, `\semiGermanChords`, `\italianChords` et `\frenchChords` permettent de définir la langue, comme vous pouvez le constater :

default	E/D	Cm	B/B	B [#] /B [#]	B ^b /B ^b
german	E/d	Cm	H/h	H [#] /his	B/b
semi-german	E/d	Cm	H/h	H [#] /his	B ^b /b
italian	Mi/Re	Do m	Si/Si	Si [#] /Si [#]	Si ^b /Si ^b
french	Mi/Ré	Do m	Si/Si	Si [#] /Si [#]	Si ^b /Si ^b



Nombre de carnets de chant allemands indiquent un accord mineur par l'emploi de caractères en minuscule, sans le suffixe m. Cette fonctionnalité est gérée par la propriété `chordNameLowercaseMinor` :

```
\chords {
  \set chordNameLowercaseMinor = ##t
  c2 d:m e:m f
}
```

C d e F

Lorsqu'aucun des systèmes prédéfinis ne vous permet de représenter l'accord voulu, les propriétés suivantes vous permettront de le construire.

`chordRootNamer`

Dans les chiffrages d'accord jazz, la note fondamentale de chaque accord est exprimée par une lettre capitale, parfois suivie d'une altération, correspondant à la notation anglo-saxonne de la musique. Cette propriété a pour valeur la fonction qui transforme la hauteur de la note fondamentale en nom de note ; c'est donc en assignant une nouvelle fonction à cette propriété que l'on peut produire des noms de note spéciaux adaptés par exemple aux systèmes de notation d'autres pays.

`majorSevenSymbol`

Cette propriété définit l'objet ajouté au `chordRootNamer` pour indiquer une septième majeure. Les options prédéfinies sont `whiteTriangleMarkup` (triangle blanc) et `blackTriangleMarkup` (triangle noir).

`additionalPitchPrefix`

Lorsqu'un chiffrage contient des notes ajoutées, vous pouvez le préfixer d'une annotation. LilyPond n'en ajoute pas par défaut, dans le but de ne pas trop surcharger la partition ; vous pouvez cependant les faire apparaître si elles sont visuellement efficaces.

```
\new ChordNames {
  <c e g d'> % add9
  \set additionalPitchPrefix = #"add"
  <c e g d'> % add9
}
```

C⁹ C^{add9}

`chordNoteNamer`

Lorsqu'un chiffrage mentionne une note ajoutée (par exemple la basse), les règles utilisées sont par défaut celles définies par la propriété `chordRootNamer` ci-dessus. Cependant, la propriété `chordNoteNamer` permet de régler cet élément indépendamment, par exemple pour imprimer la basse en caractères minuscules.

chordNameSeparator

Les différents termes d'un chiffage jazz (par exemple les notes de l'accord et la basse) sont habituellement légèrement espacés. La propriété **chordNameSeparator** permet d'indiquer un autre séparateur. Le séparateur entre un chiffage et sa basse est géré par la propriété **slashChordSeparator**.

```
\chords {
  c4:7.9- c:7.9-/g
  \set chordNameSeparator = \markup { "/" }
  \break
  c4:7.9- c:7.9-/g
}
```

$$C^{7\flat 9} \ C^{7\flat 9}/G$$

$$C^{7/\flat 9} \ C^{7/\flat 9}/G$$
slashChordSeparator

La note basse d'un accord n'est pas forcément la fondamentale. L'accord est alors « renversé » – certains diront « barré » parce que son chiffage est habituellement flanqué d'une barre oblique entre l'accord de base et sa basse. La propriété **slashChordSeparator** permet de modifier ce séparateur – par défaut la barre de fraction.

```
\chords {
  c4:7.9- c:7.9-/g
  \set slashChordSeparator = \markup { " over " }
  \break
  c4:7.9- c:7.9-/g
}
```

$$C^{7\flat 9} \ C^{7\flat 9}/G$$

$$C^{7\flat 9} \ C^{7\flat 9} \text{ over } G$$
chordNameExceptions

Cette propriété recense, sous forme de paire, les accords mis en forme de manière particulière. Le premier élément de chacune des paires répertorie les différentes hauteurs qui constituent l'accord. Le second élément est un *markup* qui sera ajouté au **chordRootNamer** lors de l'impression du chiffage.

minorChordModifier

Les accords mineurs sont habituellement identifiés par un **m** après leur fondamentale. Certaines nomenclatures ont cependant adopté un autre suffixe, comme le signe moins.

```
\chords {
  c4:min f:min7
  \set minorChordModifier = \markup { "-" }
  \break
  c4:min f:min7
}
```

Cm Fm⁷C- F-⁷**chordPrefixSpacer**

Le modificateur pour accord mineur, géré par la propriété `minorChordModifier`, est en principe accolé la fondamentale. Vous pouvez cependant l'espacer de la fondamentale à l'aide de la propriété `chordPrefixSpacer`. Notez bien que cet espacement sera réduit à néant si la fondamentale est altérée.

Commandes prédéfinies

`\whiteTriangleMarkup`, `\blackTriangleMarkup`, `\germanChords`, `\semiGermanChords`, `\italianChords`, `\frenchChords`.

Morceaux choisis*Modèles de chiffrage d'accords*

Il est possible de créer votre propre modèle de chiffrages en réglant la propriété `chordNameExceptions`.

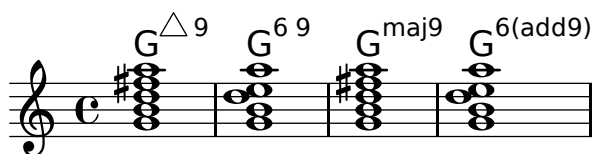
```
% modify maj9 and 6(add9)
% Exception music is chords with markups
chExceptionMusic = {
  <c e g b d'>1-\markup { \super "maj9" }
  <c e g a d'>1-\markup { \super "6(add9)" }
}

% Convert music to list and prepend to existing exceptions.
chExceptions = #( append
  ( sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

\layout {
  ragged-right = ##t
}

<< \context ChordNames \theMusic
  \context Voice \theMusic
>>
```

*Chiffrage d'un maj7*

La représentation d'un accord de septième majeure se gère par le `majorSevenSymbol`.

```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```

$C^{\triangle} C^{j7}$

Chiffrages et barres de mesure

L'ajout du graveur `Bar_engraver` à un contexte `ChordNames` permet d'imprimer les barres de mesure entre les chiffres.

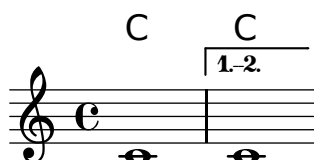
```
\new ChordNames \with {
  \override BarLine #'bar-extent = #'(-2 . 2)
  \consists "Bar_engraver"
}
\chordmode {
  f1:maj7 f:7 bes:7
}
```

$F^{\triangle} \mid F^7 \mid B^{\flat 7} \mid$

Crochet de reprise sous les chiffres d'accord

L'ajout du `Volta_engraver` à la bonne portée permet d'imprimer les crochets de reprise entre les chiffres et la portée.

```
\score {
  <<
    \chords {
      c1
      c1
    }
    \new Staff \with {
      \consists "Volta_engraver"
    }
    {
      \repeat volta 2 { c'1 }
      \alternative { c' }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



Personnalisation du séparateur d'accords

Le séparateur de termes d'un chiffage peut adopter n'importe quelle forme à l'aide d'un *markup*.

```
\chords {
  c:7sus4
  \set chordNameSeparator
    = \markup { \typewriter | }
  c:7sus4
}
```

$C^7 \text{ sus4}$ $C^7 | \text{ sus4}$

Voir aussi

Manuel de notation : Section A.1 [Table des noms d'accord], page 605, Section A.2 [Modificateurs d'accord usuels], page 606.

Essai sur la gravure musicale automatisée : Section “Références bibliographiques” dans *Essai*.

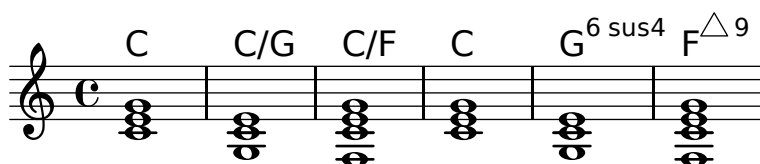
Fichiers d'initialisation : ‘scm/chords-ignatzek.scm’, ‘scm/chord-entry.scm’, ‘ly/chord-modifier-init.ly’.

Morceaux choisis : Section “Accords” dans *Morceaux choisis*.

Problèmes connus et avertissements

Les chiffrages d'accords ne sont déterminés que par la succession des hauteurs de notes. En d'autres termes, les accords inversés ne sont pas reconnus, ni les notes ajoutées à la basse. C'est pourquoi les accords saisis au moyen de la syntaxe <...> peuvent produire des chiffrages étranges.

```
myChords = \relative c' {
  \chordmode { c1 c/g c/f }
  <c e g>1 <g c e> <f c' e g>
}
<<
  \new ChordNames { \myChords }
  \new Staff { \myChords }
>>
```



2.7.3 Basse chiffrée

Adagio.

Violino I.

Violino II.

Violone,
e Cembalo.

6 # 6 6 6 #

5 6 4 5 5 6 6 5 b

6 # 6 6 5 4 b 6 6b 6 5 4b 3 # 7 6 5 9 8 4 3

LilyPond permet de générer des parties de continuo.

Introduction à la basse chiffrée

Les parties de basse continue étaient très répandues dans la musique baroque et jusqu'à la fin du XVIII^e siècle. Comme son nom l'indique, le *continuo* constitue une partie à lui seul, qui se déroule tout au long de l'œuvre pour en donner la structure harmonique.

Les musiciens du *continuo* jouent des claviers (clavecin, orgue) ou de tout autre instrument pouvant réaliser des accords. Leur partie est constituée d'une portée de basse aux notes agrémentées de combinaisons de chiffres et signes indiquant le développement des accords à jouer, ainsi que leur éventuel renversement. Cette notation était avant tout un guide, invitant le musicien à improviser de lui-même l'accompagnement.

LilyPond gère la basse chiffrée.

```
<<
\new Voice { \clef bass dis4 c d ais g fis}
\new FiguredBass {
  \figuremode {
    < 6 >4 < 7\+ >8 < 6+ [_!] >
    < 6 >4 <6 5 [3+] >
    < _ >4 < 6 5/>4
  }
}
>>
```



La gestion de la basse chiffrée se décompose en deux parties. Dans un premier temps, le mode `\figuremode` permet de saisir les accords sous forme chiffrée. Le contexte `FiguredBass` s'occupera ensuite de gérer les objets `BassFigure`. La basse chiffrée pourra être attachée à un contexte `Staff`.

L'expression `\figures{ ... }` constitue un raccourci à `\new FiguredBass { \figuremode { ... } }`.

Bien que la gestion de la basse chiffrée ressemble beaucoup à celle des accords, elle est beaucoup plus simpliste. Le mode `\figuremode` ne fait que stocker des chiffres que le contexte `FiguredBass` se chargera d'imprimer tels quels. En aucune manière ils ne sont transformés en son, et ils ne sont pas rendus dans un fichier MIDI.

Voir aussi

Glossaire musicologique : [Section “basse chiffrée” dans *Glossaire*](#).

Morceaux choisis : [Section “Accords” dans *Morceaux choisis*](#).

Saisie de la basse chiffrée

`\figuremode` permet de faire la relation entre ce qui est saisi et le mode de chiffrage. De plus amples informations quant aux différents modes sont regroupées à la rubrique [Section 5.4.1 \[Modes de saisie\], page 577](#).

En mode de saisie, un chiffrage est délimité par `<` et `>`. La durée est indiquée après le `>` :

```
\new FiguredBass {
  \figuremode {
    <6 4>2
  }
}
```

6
4

Une altération – y compris un bémol – peut s'attacher à un chiffre :

```
\figures {
  <7! 6+ 4-> <5++> <3-->
}
```

$\flat 7 \times 5 \flat 3$
 $\sharp 6$
 $\flat 4$

Vous pouvez stipuler un intervalle augmenté ou diminué :

```
\figures {
  <6\+ 5/> <7/>
}
```

$\sharp 6$ **$\flat 7$**

Vous pouvez barrer un chiffre d'une oblique inversée, typiquement pour une « petite sixte » :

```
\figures {
  <6> <6\\>
}
```

6 **$\flat 6$**

Vous pouvez insérer un espace ou ajouter des crochets :

```
\figures {
  <[12 _!] 8 [6 4]>
}
```

$\left[\begin{array}{c} 12 \\ \flat \\ 8 \\ \flat \\ 4 \end{array} \right]$

Vous pouvez aussi ajouter des chaînes de caractères ou des étiquettes – cf. [\[Text markup commands\]](#), page [\[undefined\]](#).

```
\figures {
  <\markup { \tiny \number 6 \super (1) } 5>
}
```

$6^{(1)}$
 5

Lorsque des chiffrages se répètent, vous pouvez utiliser des lignes de prolongation.

```
<<
{
  \clef bass
  e4 d c b,
  e4 d c b,
}
\figures {
  \bassFigureExtendersOn
  <6 4>4 <6 3> <7 3> <7 3>
  \bassFigureExtendersOff
  <6 4>4 <6 3> <7 3> <7 3>
}
>>
```



En pareil cas, la ligne de prolongation masquera toujours le chiffre qu'elle rappelle dans le chiffrage suivant à moins d'avoir été explicitement interrompue.

```
<<
\figures {
  \bassFigureExtendersOn
  <6 4>4 <6 4> <6\! 4\!> <6 4>
}
{
  \clef bass
  d4 d c c
}
>>
```



Voici, de manière synthétique, les différents modificateurs disponibles :

ModificateurUtilisation

Exemple

+, -, ! Altérations

$\flat 7 \times 5 \sharp 3$
 $\sharp 6$
 $\flat 4$

\+, / Augmentation ou diminution d'un degré

$+6$ 7
 5

\\ Petite sixte

$\mathfrak{6}$

\! Terminaison d'une prolongation



Commandes prédéfinies

\bassFigureExtendersOn, \bassFigureExtendersOff.

Morceaux choisis

Emplacement des altération en basse continue

On peut choisir d'imprimer les altérations et signes plus aussi bien avant qu'après les chiffres, en réglant les propriétés `figuredBassAlterationDirection` et `figuredBassPlusDirection`.

```
\figures {
  <6\+> <5+> <6 4-> r
```

```

\set figuredBassAlterationDirection = #RIGHT
<6\+> <5+> <6 4-> r
\set figuredBassPlusDirection = #RIGHT
<6\+> <5+> <6 4-> r
\set figuredBassAlterationDirection = #LEFT
<6\+> <5+> <6 4-> r
}

```

+6 #5 **6** +6 5# **6** 6+ 5# **6** 6+ #5 **6**
 ♭4 ♭4 ♭4 ♭4

Voir aussi

Morceaux choisis : [Section “Accords”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “BassFigure”](#) dans *Référence des propriétés internes*, [Section “BassFigureAlignment”](#) dans *Référence des propriétés internes*, [Section “BassFigureLine”](#) dans *Référence des propriétés internes*, [Section “BassFigureBracket”](#) dans *Référence des propriétés internes*, [Section “BassFigureContinuation”](#) dans *Référence des propriétés internes*, [Section “FiguredBass”](#) dans *Référence des propriétés internes*.

Gravure de la basse chiffrée

Une ligne de basse chiffrée s’imprime soit dans un contexte `FiguredBass`, soit dans la plupart des autres contextes du niveau de la portée.

Le contexte `FiguredBass` ne tient aucun compte des notes qui apparaissent sur la portée.

```

<<
\relative c'' {
  c4 c'8 r8 c,4 c'
}
\new FiguredBass {
  \figuremode {
    <4>4 <10 6>8 s8
    <6 4>4 <6 4>
  }
}
>>

```



Il est impératif, dans cet exemple, d’instancier explicitement le contexte `FiguredBass` pour éviter l’apparition d’une portée supplémentaire vide.

On peut ajouter une basse chiffrée directement à un contexte `Staff`. L’alignement vertical est alors automatiquement ajusté.

```

<<
\new Staff = myStaff
\figuremode {
  <4>4 <10 6>8 s8
  <6 4>4 <6 4>
}

```

```

%% Put notes on same Staff as figures
\context Staff = myStaff
{
  \clef bass
  c4 c'8 r8 c4 c'
}
>>

```

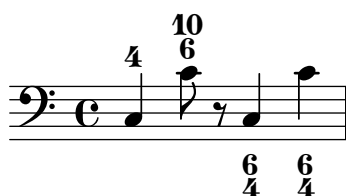


La basse chiffrée attachée à un contexte `Staff` peut se positionner au-dessus ou en dessous de la portée.

```

<<
\new Staff = myStaff
\figuremode {
  <4>4 <10 6>8 s8
  \bassFigureStaffAlignmentDown
  <6 4>4 <6 4>
}
%% Put notes on same Staff as figures
\context Staff = myStaff
{
  \clef bass
  c4 c'8 r8 c4 c'
}
>>

```



Commandes prédéfinies

`\bassFigureStaffAlignmentDown,`
`\bassFigureStaffAlignmentNeutral.`

`\bassFigureStaffAlignmentUp,`

Voir aussi

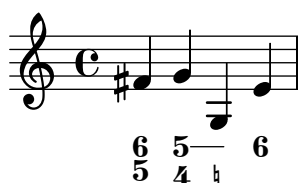
Morceaux choisis : [Section “Accords”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “BassFigure”](#) dans *Référence des propriétés internes*, [Section “BassFigureAlignment”](#) dans *Référence des propriétés internes*, [Section “BassFigureLine”](#) dans *Référence des propriétés internes*, [Section “BassFigureBracket”](#) dans *Référence des propriétés internes*, [Section “BassFigureContinuation”](#) dans *Référence des propriétés internes*, [Section “FiguredBass”](#) dans *Référence des propriétés internes*.



Ce problème peut être contourné en activant les prolongateurs derrière le chiffrage qui l'entame, puis en les désactivant derrière leur fin.

```
<<
{ fis4 g g, e' }
\figures {
  <6 5>4 <5 4>
  \bassFigureExtendersOn
  < 5 _!>4 <6>
  \bassFigureExtendersOff
}
>>
```



2.8 Musique contemporaine

L'aube du XXe siècle a vu bourgeonner nombre de techniques et de styles de composition. Qu'il s'agisse des nouveaux développements autour de l'harmonie et du rythme, de l'expansion du spectre des hauteurs et de l'évolution de nombreuses techniques instrumentales, tous ces différents phénomènes ont participé à l'évolution de la notation musicale. Les paragraphes qui suivent sont là pour vous proposer des références et informations quant à ces nouvelles techniques de notation.

2.8.1 Hauteur et harmonie en musique contemporaine

Intéressons-nous tout d'abord à ce qui relève de la notation des hauteurs et à l'harmonie en musique contemporaine.

Généralités en matière de hauteur et d'harmonie

- La notation habituelle des quarts de tons est abordée à la rubrique [Nom des notes dans d'autres langues], page 7.
- Les tonalités inhabituelles sont abordées à la rubrique [Armure], page 19.
- Les pratiques contemporaines en matière d'altération sont abordées à la rubrique [Altérations accidentelles automatiques], page 23.

Notation microtonale

Armures contemporaines et harmonie

2.8.2 Approches du rythme en musique contemporaine

Abordons à présent certaines particularités de la notation du rythme en musique contemporaine.

Généralités sur le rythme en musique contemporaine

- Les métriques composites sont abordées à la rubrique [Métrique], page 58.

- Les bases de la polymétrie sont abordées à la rubrique [Notation polymétrique], page 69.
- Certaines particularités en matière de ligature sont abordées à la rubrique [Liens de croches en soufflet], page 89.
- Les lignes de mensuration (barres de mesures uniquement entre les portées) sont abordées à la rubrique [Regroupement de portées], page 172.

Nolets et musique contemporaine

Métriques contemporaines

Notation polymétrique étendue

Ligatures et musique contemporaine

Barres de mesure et musique contemporaine

2.8.3 Notation graphique

2.8.4 Techniques de partition contemporaine

2.8.5 Nouvelles techniques instrumentales

2.8.6 Informations complémentaires et exemples pertinents

Vous trouverez ici une sélection d'ouvrages de référence, d'exemples et autres ressources qui vous permettront d'étudier plus avant la notation contemporaine.

Ouvrages et articles sur la notation en musique contemporaine

- *Music Notation in the Twentieth Century: A Practical Guidebook* par Kurt Stone [W. W. Norton, 1980]
- *Music Notation: A Manual of Modern Practice* par Gardner Read [Taplinger, 1979]
- *Instrumentation and Orchestration* par Alfred Blatter [Schirmer, 2de ed. 1997]

Partitions et exemples

2.9 Notations anciennes

Sal- ve, Re- gí- na, ma- ter mi- se- ri- cór- di- ae: Ad

te cla- má- mus, éx- su- les, fi- li- i He- vae. Ad te su- spi-

rá- mus, ge- mén- tes et flen- tes in hac la- cri-

má- rum val- le. E- ia er- go, Ad- vo- cá- ta no- stra, il-



La gestion par LilyPond des formes de notation ancienne inclut des fonctionnalités spécifiques à la notation mensurale, au chant grégorien et à la notation de style kievien. Ces fonctionnalités sont accessibles en modifiant les propriétés de style des objets graphiques tels que tête de note ou silence, ou bien grâce aux contextes prédéfinis à cet effet.

De nombreux objets graphiques – « grobs » dans le jargon de LilyPond – disposent d’une propriété `style`. Manipuler cette propriété permet d’adapter l’aspect typographique des *grobs* à une forme de notation particulière, ce qui évite la création de nouveaux concepts de notation. Voir à ce sujet

- [Têtes de note anciennes], page 414,
- [Altérations et armures anciennes], page 417,
- [Silences anciens], page 416,
- [Clefs anciennes], page 412,
- [Clefs grégoriennes], page 420,
- [Crochets anciens], page 415,
- [Métriques anciennes], page 413.

D’autres aspects de la notation ancienne ne peuvent pas être gérés aussi simplement qu’en jouant sur les propriétés d’un style appliqué à un objet graphique ou en lui ajoutant des articulations. Certains concepts sont spécifiques à la notation ancienne.

- [Guidons], page 411,
- [Divisions], page 421,
- [Ligatures], page 410.

Voir aussi

Glossaire musicologique : Section “custos” dans *Glossaire*, Section “ligature” dans *Glossaire*, Section “mensural notation” dans *Glossaire*.

Manuel de notation : [Têtes de note anciennes], page 414, [Altérations et armures anciennes], page 417, [Silences anciens], page 416, [Clefs grégoriennes], page 420, [Crochets anciens], page 415, [Métriques anciennes], page 413, [Guidons], page 411, [Divisions], page 421, [Ligatures], page 410.

Formes de notation ancienne prises en charge

En matière de chant grégorien, LilyPond dispose de trois différents styles :

- *Editio Vaticana* constitue un style à part entière dédié au chant grégorien, le plus approchant possible des éditions de Solesmes, éditeur officiel du Vatican depuis 1904. LilyPond dispose de tous les signes de notation propres à ce style, y compris les ligatures, custodes et certaines spécificités comme le quilisma et l'oriscus.
- *Editio Medicaea* dispose d'un certain nombre de spécificités des éditions qui faisaient autorité avant Solesmes. On le connaît aussi sous le nom de Ratisbone. Ce qui le distingue le plus du style *Vaticana* réside dans les clefs, en forme de barres obliques, et les têtes de note, plus carrées et régulières.
- Le style *Hufnagel* (« clou de fer à cheval ») ou *gothique* imite le style des manuscrits médiévaux d'Allemagne et d'Europe centrale. Il tire son nom de l'allure des notes (en virgule ou *virga*) qui ressemblent à des têtes de clou.

LilyPond dispose de trois styles imitant les manuscrits du Bas Moyen Âge et de la Renaissance ainsi que les premières impressions de musique mesurée.

- Le style *Mensural* est celui qui se rapproche le plus des manuscrits de la période allant de la fin du Moyen Âge au début de la Renaissance, avec ses petites têtes de note en forme de losange étroit et ses silences comme dessinés à main levée.
- Le style *Neomensural* est une version moderne et stylisée du style mensural : les têtes de note sont un peu plus galbées et les silences plus rectilignes. Ce style est tout à fait approprié à l'incipit d'une transcription de musique ancienne.
- Le style *Petrucchi* tire son nom du fameux graveur vénitien Ottaviano Petrucci (1466-1539), premier imprimeur à utiliser des caractères amovibles pour la musique dans son édition du *Harmonice musices odhecaton* en 1501. Les têtes de notes de ce style sont plus larges que pour les autres styles mensuraux.

Bien qu'il ne soient pas complets, les styles *Baroque* et *Classical* diffèrent du style par défaut par quelques détails – certaines têtes de note pour le *Baroque* et le soupir pour le *Classical*.

Seul le style mensural dispose de signes alternatifs couvrant tous les aspects de la notation. Ainsi, les silences et les crochets sont absents du style grégorien puisqu'ils ne sont pas utilisés dans la notation du plain-chant ; le style Petrucci ne dispose en propre d'aucun crochet ni d'altération.

Chacun des éléments de notation peut donc être modifié de manière indépendante jusqu'à, pourquoi pas, utiliser dans une même partition des crochets en *Mensural*, des têtes de note de *Petrucchi*, des silences du *Classical* et des clefs du style *Vaticana*.

Voir aussi

Glossaire musicologique : [Section “mensural notation”](#) dans *Glossaire*, [Section “flag”](#) dans *Glossaire*.

2.9.1 Considérations communes aux musiques anciennes

Contextes prédéfinis

LilyPond dispose, en matière de musique ancienne et de grégorien, de contextes prédéfinis. Ceux-ci contiennent tout ce qui est nécessaire à la gestion d'une voix ou d'une portée selon le style adopté. Si tout cela vous dépasse et que vous désirez plonger dans le vif du sujet sans trop vous préoccuper d'ajuster des contextes, consultez les pages dédiées aux contextes prédéfinis. Ils vous permettront d'adapter vos contextes de voix et de portée, et vous n'aurez plus qu'à saisir les notes dans un contexte `VaticanaVoice`, `VaticanaStaff`, `MensuralVoice` ou `MensuralStaff`. Vous trouverez des détails sur ces contextes aux rubriques

- [Contextes du chant grégorien], page 419,
- [Contextes de musique mensurale], page 411.

Voir aussi

Glossaire musicologique : [Section “mensural notation” dans *Glossaire*](#).

Manuel de notation : [\[Contextes de musique mensurale\]](#), page 411, [\[Contextes du chant grégorien\]](#), page 419.

Ligatures

Une ligature est un symbole graphique qui représente un groupe d’au moins deux notes distinctes. Les ligatures ont commencé à apparaître dans les manuscrits de chant grégorien, pour indiquer des suites ascendantes ou descendantes de notes.

Les ligatures s’indiquent par un bornage entre `\[` et `\]`. Certains styles de ligature peuvent demander un complément de syntaxe spécifique. Par défaut, le graveur `LigatureBracket` place un simple crochet au dessus de la ligature :

```
\transpose c c' {
  \[ g c a f d' \]
  a g f
  \[ e f a g \]
}
```



Deux autres styles de ligature sont accessibles : `vatican` pour le grégorien et `mensural` pour la musique ancienne (seules sont disponibles les ligatures mensurales blanches, avec quelques limitations). Selon le style de ligature désiré, il faut remplacer, dans le contexte `Voice` approprié, le graveur `Ligature_bracket_engraver` par le graveur de ligature qui convient – voir les rubriques [\[Ligatures mensurales\]](#), page 418 et [\[Neumes et ligatures grégoriennes\]](#), page 423 à ce sujet.

Voir aussi

Glossaire musicologique : [Section “ligature” dans *Glossaire*](#).

Manuel de notation : [\[Ligatures mensurales\]](#), page 418, [\[Neumes et ligatures grégoriennes\]](#), page 423.

Problèmes connus et avertissements

La gestion de l’espacement spécifique aux ligatures n’est à ce jour pas implémentée. En conséquence, les ligatures sont trop espacées les unes des autres et les sauts de ligne mal ajustés. De plus, les paroles ne s’alignent pas de manière satisfaisante en présence de ligatures.

Les altérations ne pouvant être imprimées à l’intérieur d’une ligature, il faut les rassembler et les imprimer juste avant.

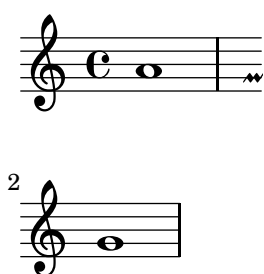
La syntaxe utilisée correspond à l’ancienne convention de préfixage `\[expression musicale\]`. Pour des raisons d’uniformité, nous opterons probablement pour le style en suffixe (postfix) `note\[... note\]`.

Guidons

Un guidon — *custos*, pluriel *custodes* en latin — est un symbole qui apparaît à la fin d’une portée. Il montre la hauteur de la ou des premières notes de la portée suivante, donnant une indication judicieuse à l’exécutant.

Les guidons étaient couramment utilisés jusqu’au XVII^e siècle. De nos jours, on les retrouve uniquement dans quelques formes particulières de notation telles que les éditions contemporaines de chant grégorien comme les *editio vaticana*. Différents glyphes existent selon le style de notation.

L’impression de guidons s’obtient en affectant, dans un bloc `\layout`, le [Section “Custos-engraver”](#) dans [Référence des propriétés internes](#) au contexte `Staff`, comme le montre l’exemple suivant.



Le glyphe du guidon est déterminé par la propriété `style`. Les styles disponibles sont `vaticana`, `medicaea`, `hufnagel` et `mensural`. En voici un aperçu :

vaticana	medicaea	hufnagel	mensural
↓	↓	✓	↘

Voir aussi

Glossaire musicologique : [Section “custos”](#) dans *Glossaire*.

Référence des propriétés internes : [Section “Custos”](#) dans *Référence des propriétés internes*.

Morceaux choisis : [Section “Musiques anciennes”](#) dans *Morceaux choisis*.

Prise en charge de la basse figurée

LilyPond gère partiellement la représentation de basses chiffrées, typiques de l’époque baroque, mais également employées de nos jours en harmonie et en analyse – voir la rubrique [Section 2.7.3 \[Basse chiffrée\]](#), page 399.

Voir aussi

Glossaire musicologique : [Section “Basse figurée”](#) dans *Glossaire*.

Manuel de notation : [Section 2.7.3 \[Basse chiffrée\]](#), page 399.

2.9.2 Typographie de musique ancienne

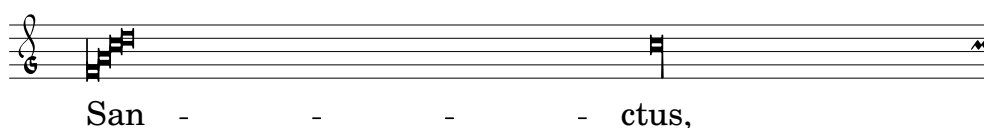
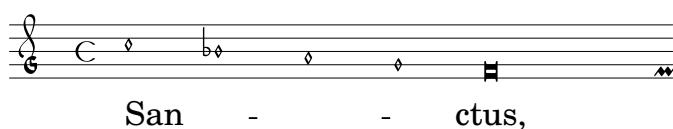
Contextes de musique mensurale

Les contextes `MensuralVoice` et `MensuralStaff` permettent de graver des chants dans le style mesuré. Ces contextes initialisent les propriétés de tous les autres contextes et objets graphiques à des valeurs adéquates, de telle sorte que vous pouvez tout de suite vous lancer dans la saisie de votre chant comme ci-après :

```

\score {
  <<
    \new MensuralVoice = "discantus" \transpose c c' {
      \override Score.BarNumber #'transparent = ##t {
        c'1\melisma bes a g\melismaEnd
        f\breve
        \[ f1\melisma a c'\breve d'\melismaEnd \]
        c'\longa
        c'\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}

```



Voir aussi


Glossaire musicologique : [Section “mensural notation” dans *Glossaire*](#).







Clefs anciennes

Avec LilyPond, de nombreuses clés sont disponibles, dont certaines sont dédiées à la musique ancienne.

Le tableau suivant présente les différentes clés anciennes que vous pouvez sélectionner avec la commande `\clef`. Certaines de ces clés utilisent le même glyphe, attaché à l’une ou l’autre des lignes de la portée. Le chiffre porté en suffixe permet alors de les différencier. Vous pouvez néanmoins forcer le positionnement du glyphe sur une ligne, comme expliqué à la section [\[Clefs\]](#), [page 16](#). Dans la colonne exemple, la note suivant la clé est un do médium.

Les clés d’ut de Petrucci avaient une hampe gauche différente selon leur ligne de rattachement.

Description	Clef disponible	Exemple
Clef d’ut, style mensural historique	<code>mensural-c1</code> , <code>mensural-c2</code> , <code>mensural-c3</code> , <code>mensural-c4</code>	

Clef de fa, style mensural historique	<code>mensural-f</code>	
Clef de sol, style mensural historique	<code>mensural-g</code>	
Clef d'ut, style mensural moderne	<code>neomensural-c1</code> , <code>neomensural-c2</code> , <code>neomensural-c3</code> , <code>neomensural-c4</code>	
Clef d'ut, style mensural Petrucci, positionnable sur différentes lignes (clef d'ut seconde pour l'exemple)	<code>petrucci-c1</code> , <code>petrucci-c2</code> , <code>petrucci-c3</code> , <code>petrucci-c4</code> , <code>petrucci-c5</code>	
Clef de fa, style mensural Petrucci, positionnable sur différentes lignes (clef de fa troisième pour l'exemple)	<code>petrucci-f3</code> , <code>petrucci-f4</code> , <code>petrucci-f5</code>	
Clef de sol, style mensural Petrucci	<code>petrucci-g</code>	

Moderne signifie « gravé comme dans les transcriptions contemporaines de musique mesurée. »

Petrucci signifie « inspiré des éditions réalisées par le maître graveur Petrucci (1466-1539). »

Historique signifie « gravé comme dans les éditions historiques, manuscrites ou non, autres que celles de Petrucci. »

Voir aussi

Glossaire musicologique : [Section “mensural notation”](#) dans *Glossaire*, [Section “clef”](#) dans *Glossaire*.

Manuel de notation : [\[Clefs\]](#), [page 16](#).

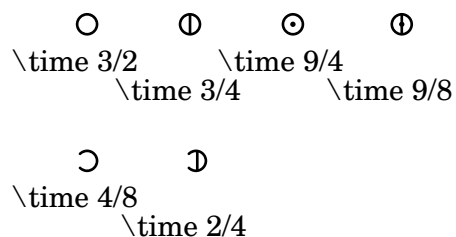
Problèmes connus et avertissements

La clef de sol mensurale est calquée sur celle de Petrucci.

Métriques anciennes

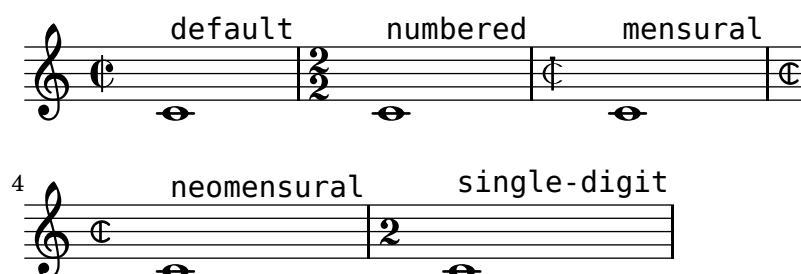
Les chiffrages de métrique mensurale sont partiellement pris en charge. Les glyphes ne font que représenter des métriques particulières. En d'autres termes, pour obtenir le glyphe correspondant à une métrique mensurale particulière à l'aide de la commande `\time n/m`, vous devez choisir la paire (n,m) parmi les valeurs suivantes :

\mathbb{C}	\mathbb{C}	\mathbb{C}	\mathbb{C}
<code>\time 4/4</code>	<code>\time 6/4</code>	<code>\time 2/2</code>	<code>\time 6/8</code>



La propriété `style` de l'objet `TimeSignature` permet d'accéder aux indicateurs de métrique anciens. Les styles `neomensural` et `mensural` sont disponibles. Vous avez vu ci-dessus le style `neomensural`, particulièrement utilisé pour l'incipit des transcriptions. Le style `mensural` imite l'aspect de certaines éditions du XVI^e siècle.

Voici les différences entre les styles :



La rubrique [Métrique], page 58 expose les principes généraux sur l'utilisation des indications de métrique.

Voir aussi

Glossaire musicologique : [Section “mensural notation” dans *Glossaire*](#).

Manuel de notation : [Métrique], page 58.

Problèmes connus et avertissements

Les équivalences de durées de note ne sont pas modifiées par un changement de métrique. Par exemple, l'équivalence une brève pour trois semi-brèves (*tempus perfectum*) doit s'effectuer à la main en entrant :

```
breveTP = #(ly:make-duration -1 0 3 2)
...
{ c\breveTP f1 }
```

Ce qui définira `breveTP` à $3/2$ fois $2 = 3$ fois une ronde.

Les symboles `mensural68alt` et `neomensural68alt` – alternatives à la métrique $6/8$ – ne sont pas accessibles par la commande `\time`. Utilisez alors un `\markup {\musicglyph #"timesig.mensural68alt" }`.

Têtes de note anciennes

Pour de la musique ancienne, vous disposez de plusieurs styles de tête de note, en plus du style par défaut `default`. Vous pouvez affecter à la propriété `style` de l'objet `NoteHead` les valeurs `baroque`, `neomensural`, `mensural`, `petrucci`, `blackpetrucci` ou `semipetrucci`.

Le style `baroque` diffère du style `default` par

- la disponibilité de la `maxima`, et
- la `\breve` qui sera carrée et non pas ovoïde.

Les styles `neomensural`, `mensural` et `petrucci` diffèrent du `baroque` par ceci :

- les notes de durée inférieure ou égale à une ronde sont en forme de losange, et
- les hampes sont centrées sur la tête.

Le style `blackpetrucci` permet d'obtenir, en notation mensurale blanche, des têtes noircies. Cependant, et dans la mesure où le style de tête n'influence en rien le nombre des crochets, une *semiminima* devrait alors se noter `a8*2` plutôt que `a4`, de telle sorte qu'elle ne se confonde pas avec une *minima*. Le multiplicateur peut varier, pour indiquer par exemple un triolet.

Le style `semipetrucci` permet de partiellement noircir certaines têtes, comme la brève, la longue et la maxime.

L'exemple suivant illustre le style `petrucci`.

```
\set Score.skipBars = ##t
\autoBeamOff
\override NoteHead #'style = #'petrucci
a'\maxima a'\longa a'\breve a'1 a'2 a'4 a'8 a'16 a'
\override NoteHead #'style = #'semipetrucci
a'\breve*5/6
\override NoteHead #'style = #'blackpetrucci
a'8*4/3 a'
\override NoteHead #'style = #'petrucci
a'\longa
```



La rubrique [Section 1.1.4 \[Têtes de note\]](#), page 33 présente tous les styles de notes disponibles.

Voir aussi

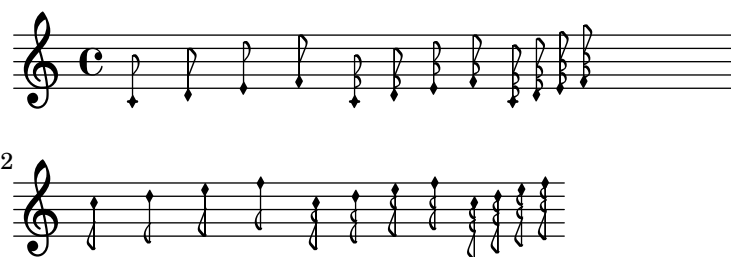
Glossaire musicologique : [Section “mensural notation”](#) dans *Glossaire*, [Section “note head”](#) dans *Glossaire*.

Manuel de notation : [Section 1.1.4 \[Têtes de note\]](#), page 33.

Crochets anciens

Le réglage de la propriété `flag-style` de l'objet hampe (`Stem`) donne accès aux crochets de style ancien. Les seuls styles actuellement pris en charge sont `default` et `mensural`.

```
\override Flag #'style = #'mensural
\override Stem #'thickness = #1.0
\override NoteHead #'style = #'mensural
\autoBeamOff
c'8 d'8 e'8 f'8 c'16 d'16 e'16 f'16 c'32 d'32 e'32 f'32 s8
c''8 d''8 e''8 f''8 c''16 d''16 e''16 f''16 c''32 d''32 e''32 f''32
```



Notez que, pour chaque crochet mensural, l'extrémité la plus proche de la tête de note sera toujours attachée à une ligne de la portée.

Il n'existe pas de crochet spécifique au style néomensural.

Les crochets n'existent pas en notation grégorienne.

Voir aussi

Glossaire musicologique : [Section “mensural notation”](#) dans *Glossaire*, [Section “flag”](#) dans *Glossaire*.

Problèmes connus et avertissements

Les crochets anciens s’attachent aux hampes avec un léger décalage.

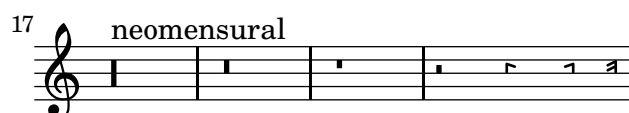
L’alignement vertical des crochets par rapport aux lignes de la portée sous-entend que les hampes se terminent toujours soit sur une ligne, soit à l’exact milieu d’un interligne. Ceci n’est pas toujours réalisable, surtout si vous faites appel à des fonctionnalités avancées de présentation de la notation classique qui, par définition, ne sont pas prévues pour être appliquées à la notation mensurale.

Silences anciens

La propriété `style` de l’objet `Rest` permet d’obtenir des silences de type ancien. Vous disposez des styles `classical`, `neomensural` et `mensural`. Le style `classical` ne se distingue du style `default` que par le soupir (demi-soupir en miroir). Le style `neomensural` convient tout à fait à l’incipit lors de la transcription de musique mensurale. Le style `mensural`, enfin, imite la gravure des silences dans certaines éditions du XVI^e siècle.

L’exemple suivant illustre les styles `mensural` et `neomensural`.

```
\set Score.skipBars = ##t
\override Rest #'style = #'classical
r\longa^"classical" r\breve r1 r2 r4 r8 r16 s \break
\override Rest #'style = #'mensural
r\longa^"mensural" r\breve r1 r2 r4 r8 r16 s \break
\override Rest #'style = #'neomensural
r\longa^"neomensural" r\breve r1 r2 r4 r8 r16
```



Les styles `mensural` et `neomensural` ne disposent pas des huitième et seizième de soupir ; LilyPond utilise dans de tels cas le style par défaut.

Voir aussi

Glossaire musicologique : [Section “mensural notation”](#) dans *Glossaire*.

Manuel de notation : [\[Silences\]](#), page 51.

Morceaux choisis : [Section “Notations anciennes”](#) dans *Morceaux choisis*.

Problèmes connus et avertissements

En style mensural, LilyPond utilise pour un silence correspondant à une *maxima*, le même glyphe que pour un *longa* ; il faut donc les multiplier pour obtenir la durée *ad hoc*. Des silences correspondant à une *longa* ne sont pas groupés automatiquement ; utilisez en pareil cas des « notes silencieuses ».

Altérations et armures anciennes

Le style `mensural` dispose d'un dièse et d'un bémol différents du style par défaut ; en cas de besoin, le bécarré sera emprunté au style `vaticana`.

mensural

♭ ✕

Pour disposer des formes anciennes d'altération, utilisez la propriété `glyph-name-alist` des objets graphiques `Accidental` et `KeySignature`, comme ceci :

```
\override Staff.Accidental #'glyph-name-alist =
  #alteration-mensural-glyph-name-alist
```

Voir aussi

Glossaire musicologique : [Section “mensural notation” dans *Glossaire*](#), [Section “Pitch names” dans *Glossaire*](#), [Section “accidental” dans *Glossaire*](#), [Section “key signature” dans *Glossaire*](#).

Manuel de notation : [Section 1.1 \[Hauteurs\]](#), page 1, [\[Altérations\]](#), page 5, [\[Altérations accidentelles automatiques\]](#), page 23, [\[Armure\]](#), page 19.

Référence des propriétés internes : [Section “KeySignature” dans *Référence des propriétés internes*](#).

Altérations suggérées (*musica ficta*)

Dans la pratique ancienne, avant le XVII^e siècle, les altérations accidentelles de l'échelle modale n'étaient pas systématiquement notées et il incombait aux chanteurs, en fonction de certaines règles, de décider s'ils devaient chanter tel degré bémol, bécarré ou dièse. Cette technique est appelée *musica ficta*. Les transcriptions modernes de telles œuvres font apparaître ces altérations en surplomb de la note.

La reproduction de ces altérations suggérées est assurée par l'activation de la fonction `suggestAccidentals`.

```
fis gis
\set suggestAccidentals = ##t
ais bis
```



Cette fonction considérera **toute** altération comme étant de la *musica ficta*, ce tant qu'elle n'aura pas été désactivée par un `\set suggestAccidentals = ##f`. Il est de ce fait plus pratique de recourir à une clause `\once \set suggestAccidentals = ##t`, qui peut tout à fait faire l'objet d'un raccourci :

```
ficta = { \once \set suggestAccidentals = ##t }
\score { \relative c''
  \new MensuralVoice {
```

```

\once \set suggestAccidentals = ##t
bes4 a2 g2 \ficta fis8 \ficta e! fis2 g1
}
}

```



Voir aussi

Référence des propriétés internes : [Section “Accidental_engraver”](#) dans *Référence des propriétés internes*, [Section “AccidentalSuggestion”](#) dans *Référence des propriétés internes*.

Ligatures mensurales

Les ligatures mensurales blanches sont prises en charge, avec des limitations.

La gravure des ligatures mensurales blanches s’obtient après avoir remplacé, dans le contexte Voice, le `Ligature_bracket_engraver` par le `Mensural_ligature_engraver`, comme ici :

```

\layout {
  \context {
    \Voice
    \remove "Ligature_bracket_engraver"
    \consists "Mensural_ligature_engraver"
  }
}

```

Lorsque le code ci-dessus est employé, l’aspect d’une ligature mensurale blanche est déterminé à partir des hauteurs et durées des notes qui la composent. Bien que cela demande un temps d’adaptation au nouvel utilisateur, cette méthode offre l’avantage que toute l’information musicale incluse dans la ligature est connue en interne. Ceci est non seulement important pour le rendu MIDI, mais aussi pour des questions de transcription automatisée d’une ligature.

Il se peut que deux notes consécutives puissent être représentées aussi bien par deux carrées que par un parallélogramme oblique (en forme de flexe). Par défaut, LilyPond présentera deux carrés ; l’impression d’une flexe s’obtient par affectation, pour la **deuxième** note, de la propriété `ligature-flexa`. Le réglage de la longueur d’une flexe se gère par la propriété de tête de note `flexa-width`.

Par exemple,

```

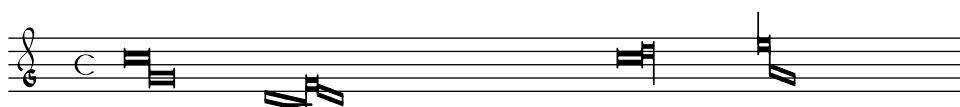
\score {
  \transpose c c' {
    \set Score.timing = ##f
    \set Score.defaultBarType = "empty"
    \override NoteHead #'style = #'petrucci
    \override Staff.TimeSignature #'style = #'mensural
    \clef "petrucci-g"
    \[ c'\maxima g \]
    \[ d\longa
      \override NoteHead #'ligature-flexa = ##t
      \once \override NoteHead #'flexa-width = #3.2
      c\breve f e d \]
    \[ c'\maxima d'\longa \]
    \[ e'1 a g\breve \]
  }
}

```

```

}
\layout {
  \context {
    \Voice
    \remove "Ligature_bracket_engraver"
    \consists "Mensural_ligature_engraver"
  }
}
}

```



Si on ne remplace pas le `Ligature_bracket_engraver` par le `Mensural_ligature_engraver`, on obtient



Voir aussi

Glossaire musicologique : [Section “ligature” dans *Glossaire*](#).

Manuel de notation : [\[Neumes et ligatures grégoriennes\]](#), page 423, [\[Ligatures\]](#), page 410.

Problèmes connus et avertissements

L’espacement horizontal n’est pas des meilleurs. Les altérations peuvent se chevaucher avec les notes précédentes.

2.9.3 Typographie du chant grégorien

Si vous écrivez en notation grégorienne, le `Vaticana_ligature_engraver` se chargera de sélectionner les têtes de note appropriées ; il est donc inutile de spécifier le style à utiliser. Vous pouvez cependant spécifier par exemple le style `vaticana_punctum` pour obtenir des neumes punctums. De même, c’est le `Mensural_ligature_engraver` qui se chargera des ligatures mensurales.

Voir aussi

Glossaire musicologique : [Section “ligature” dans *Glossaire*](#).

Manuel de notation : [\[Ligatures mensurales\]](#), page 418, [\[Ligatures\]](#), page 410.

Contextes du chant grégorien

Les contextes prédéfinis `VaticanaVoiceContext` et `VaticanaStaffContext` permettent de graver le chant grégorien dans le style des éditions vaticanes. Ces contextes initialisent les propriétés de tous les autres contextes et objets graphiques à des valeurs adéquates, de telle sorte que vous pouvez tout de suite vous lancer dans la saisie de votre chant, comme ci-dessous :

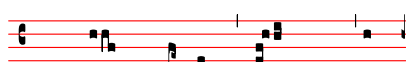
```

\include "gregorian.ly"
\score {
  <<
    \new VaticanaVoice = "cantus" {
      \[ c'\melisma c' \flexa a \]
    }
  >>
}

```

```

\[\ a \flexa \deminutum g\melismaEnd \]
f \divisioMinima
\[\ f\melisma \pes a c' c' \pes d'\melismaEnd \]
c' \divisioMinima \break
\[\ c'\melisma c' \flexa a \]
\[\ a \flexa \deminutum g\melismaEnd \] f \divisioMinima
}
\new Lyrics \lyricsto "cantus" {
  San- ctus, San- ctus, San- ctus
}
>>
}
```



San- ctus, San- ctus,



San- ctus

Clefs grégoriennes

Le tableau suivant présente les différentes clefs grégoriennes que vous pouvez sélectionner avec la commande `\clef`. Certaines de ces clefs utilisent le même glyphe, attaché à l'une ou l'autre des lignes de la portée. Le chiffre porté en suffixe permet alors de les différencier ; la numérotation des lignes va de bas en haut. Vous pouvez néanmoins forcer le positionnement du glyphe sur une ligne, comme expliqué à la section [\[Clefs\], page 16](#). Dans la colonne exemple, la note suivant la clef est un do médium.

Description	Clef disponible	Exemple
Clef d'ut, style des éditions vaticanes	<code>vaticana-do1</code> , <code>vaticana-do2</code> , <code>vaticana-do3</code>	
Clef de fa, style des éditions vaticanes	<code>vaticana-fa1</code> , <code>vaticana-fa2</code>	
Clef d'ut, style Editio Medicaea	<code>medicaea-do1</code> , <code>medicaea-do2</code> , <code>medicaea-do3</code>	
Clef de fa, style Editio Medicaea	<code>medicaea-fa1</code> , <code>medicaea-fa2</code>	
Clef d'ut, style historique Hufnagel	<code>hufnagel-do1</code> , <code>hufnagel-do2</code> , <code>hufnagel-do3</code>	

Clef de fa, style historique Hufnagel `hufnagel-fa1`, `hufnagel-fa2`



Clef combinée ut/fa, style historique Hufnagel `hufnagel-do-fa`



Voir aussi

Glossaire musicologique : [Section “clef” dans *Glossaire*](#).

Manuel de notation : [\[Clefs\]](#), page 16.

Altérations et armures grégoriennes

LilyPond dispose d’altérations pour les trois styles grégoriens :

`vaticana medicaea hufnagel`



Vous noterez que chacun de ces styles ne comporte pas toutes les altérations. LilyPond changera de style s’il est besoin d’une altération indisponible dans le style utilisé.

Pour disposer des formes anciennes d’altération, utilisez la propriété `glyph-name-alist` des objets graphiques `Accidental` et `KeySignature`, comme ceci :

```
\override Staff.Accidental #'glyph-name-alist =
  #alteration-mensural-glyph-name-alist
```

Voir aussi

Glossaire musicologique : [Section “accidental” dans *Glossaire*](#), [Section “key signature” dans *Glossaire*](#).

Manuel de notation : [Section 1.1 \[Hauteurs\]](#), page 1, [\[Altérations\]](#), page 5, [\[Altérations accidentelles automatiques\]](#), page 23, [\[Armure\]](#), page 19.

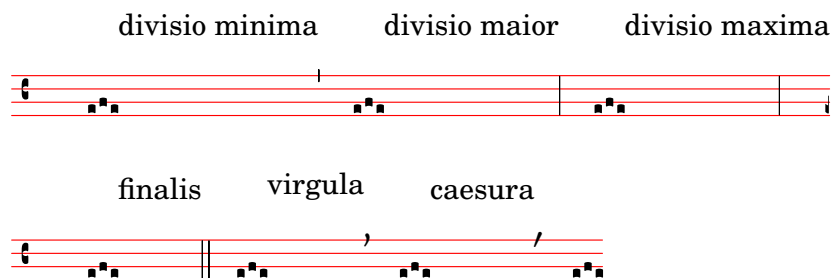
Référence des propriétés internes : [Section “KeySignature” dans *Référence des propriétés internes*](#).

Divisions

Il n’existe pas de silence en notation grégorienne. On y parle plutôt de *divisions*, *pauses* ou *coupes*.

Une division – *divisio*, pluriel *divisiones* en latin – est un symbole ajouté à la portée et utilisé en chant grégorien pour séparer les phrases ou parties. *Divisio minima*, *divisio maior* et *divisio maxima* peuvent respectivement s’interpréter comme une pause courte, moyenne ou longue, à l’image des marques de respiration — cf. [\[Signes de respiration\]](#), page 123. Le signe *finalis* n’est pas uniquement une marque de fin de chant ; il sert aussi à indiquer la fin de chaque partie dans une structure verset/répons.

Les divisions sont disponibles après inclusion du fichier ‘`gregorian-init.ly`’. Ce fichier définit les commandes `\divisioMinima`, `\divisioMaior`, `\divisioMaxima` et `\finalis`. Certaines éditions utilisent *virgula* ou *caesura* en lieu et place de *divisio minima* ; c’est pourquoi ‘`gregorian-init.ly`’ définit aussi `\virgula` et `\caesura`.



Commandes prédéfinies

`\virgula`, `\caesura`, `\divisioMinima`, `\divisioMaior`, `\divisioMaxima`, `\finalis`.

Voir aussi

Glossaire musicologique : [Section “caesura” dans *Glossaire*](#), [Section “divisio” dans *Glossaire*](#).

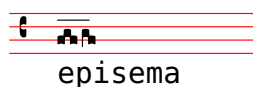
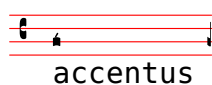
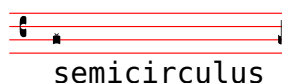
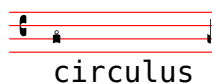
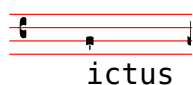
Manuel de notation : [\[Signes de respiration\]](#), page 123.

Fichiers d’initialisation : ‘`gregorian.ly`’.

Articulations grégoriennes

En plus des signes d’articulation standards décrits à la section [\[Articulations et ornements\]](#), page 109, LilyPond fournit des articulations spécifiquement destinées au style des éditions vaticanes.

```
\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \override TextScript #'font-family = #'typewriter
    \override TextScript #'font-shape = #'upright
    \override Script #'padding = #-0.1
    a\ictus_"ictus " \bar "" \break
    a\circulus_"circulus " \bar "" \break
    a\semicirculus_"semicirculus " \bar "" \break
    a\accentus_"accentus " \bar "" \break
    \[ a_"episema" \epistemInitium \pes b \flexa a b \epistemFinis \flexa a \]
  }
}
```



Voir aussi

Manuel de notation : [Articulations et ornements], page 109.

Morceaux choisis : Section “Musiques anciennes” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Episema” dans *Référence des propriétés internes*, Section “EpisemaEvent” dans *Référence des propriétés internes*, Section “Episema_engraver” dans *Référence des propriétés internes*, Section “Script” dans *Référence des propriétés internes*, Section “ScriptEvent” dans *Référence des propriétés internes*, Section “Script_engraver” dans *Référence des propriétés internes*.

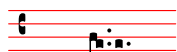
Problèmes connus et avertissements

Certaines articulations sont verticalement trop proches de leurs têtes de note.

Points d’augmentation (*morae*)

Les points d’*augmentum*, ou *morae*, s’obtiennent avec la fonction `\augmentum`. Notez que cette fonction `\augmentum` est implémentée en tant que fonction unaire plutôt que comme un préfixe de note. Par conséquent, `\augmentum \virga c` ne donnera rien de particulier. Il faut l’utiliser avec la syntaxe `\virga \augmentum c` ou `\augmentum {\virga c}`. Par ailleurs, l’expression `\augmentum {a g}` constitue une forme abrégée de `\augmentum a \augmentum g`.

```
\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \[ \augmentum a \flexa \augmentum g \]
    \augmentum g
  }
}
```



Voir aussi

Manuel de notation : [Signes de respiration], page 123.

Référence des propriétés internes : Section “BreathingSign” dans *Référence des propriétés internes*.

Morceaux choisis : Section “Musiques anciennes” dans *Morceaux choisis*.

Neumes et ligatures grégoriennes

Les neumes grégoriens conformément au style des éditions vaticanes sont pris en charge de façon assez limitée. Les ligatures élémentaires sont déjà disponibles, mais beaucoup de règles typographiques ne sont pas encore implémentées, notamment l’espacement horizontal des enchaînements de ligatures, l’alignement des paroles ou une gestion convenable des altérations.

La prise en charge des neumes grégoriens est activée par l’inclusion, en début de votre fichier source, du fichier d’initialisation ‘`gregorian.ly`’. Ceci aura pour effet de rendre disponible un certain nombre de commandes dans le but de reproduire les symboles de neumes tels qu’ils apparaissent dans la notation du plain-chant.

Les têtes de note peuvent être *modifiées* ou *jointes*.

- L’aspect d’une tête de note se modifie en *préfixant* le nom d’une hauteur par l’une des commandes suivantes : `\virga`, `\strophæ`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.

- Une ligature, autrement dit la juxtaposition de notes, s’obtient en plaçant une commande de jointure `\pes` ou `\flexa` pour marquer une ligne mélodique respectivement ascendante ou descendante, entre les notes qui la composent.

Une hauteur sans qualificatif sera considérée comme un *punctum*. Tout autre neume, y compris ceux d’une seule note d’aspect particulier comme la *virga*, sera considéré en tant que ligature et devra répondre à la syntaxe `\[...]`.

Neumes simples :

- Le *punctum* représente l’aspect standard d’une note – dans le style *Vaticana*, il s’agit d’un carré plein légèrement incurvé pour une question d’esthétique. Existe aussi le *punctum inclinatum* – carré penché qui s’obtient grâce au préfixe `\inclinatum`. Un *punctum* standard peut se modifier par le préfixe `\cavum` qui l’évidera, ou le préfixe `\linea`, qui lui adjoindra une ligne verticale de part et d’autre.
- La *virga* dispose d’une hampe sur la droite. Elle s’obtient à l’aide du modificateur `\virga`.

Ligatures

Contrairement à la majorité des autres systèmes de notation neumatique, la manière de saisir les neumes n’a rien à voir avec leur apparence typographique ; elle se concentre plutôt sur le sens musical. Ainsi, `\[a \pes b \flexa g]` produit un *torculus* constitué de trois *punctums*, alors que `\[a \flexa g \pes b]` produit un *porrectus* avec une flexe incurvée et un seul *punctum*. Il n’existe pas de commande à proprement parler qui permette de spécifier la courbe d’une flexe ; c’est la source musicale qui va le déterminer. Le fondement d’une telle approche réside dans la distinction que nous faisons entre les aspects musicaux de la source et le style de notation que nous voulons obtenir. De ce fait, la même source pourra être utilisée pour imprimer dans un autre style de notation grégorienne.

Neumes liquescents

Autre grande catégorie de notes que l’on trouve en grégorien, les neumes liquescents. Ils s’utilisent dans certaines circonstances, quand l’articulation d’une syllabe avec la suivante se fait par une « consonne liquide », sur la dernière note du groupe : M (omnis, summo), L, N, Y (ejus), NG (sanctus), W (autem, laudat). Ces consonnes ou semi-consonnes sont chantées à la hauteur correspondante, comme le seraient des voyelles, mais le chant est entravé par leur prononciation. De fait, les neumes liquescents ne sont jamais utilisés isolément (bien que rien ne l’interdise) et tombent toujours à la fin d’une ligature.

Les neumes liquescents peuvent se présenter graphiquement de deux façons différentes et relativement interchangeables : une note plus petite, ou une « bascule » verticale de la note principale. La première option s’obtient en créant un `pes` ou une `flexa` puis une modification de l’aspect de la deuxième note : `\[a \pes \deminutum b]`. La seconde option consiste à modifier l’aspect d’un neume sur note unique avec un `\auctum` tout en lui affectant une direction `\descendens` ou `\ascendens` : `\[\auctum \descendens a]`.

Signes spéciaux

Une troisième catégorie regroupe quelques signes dont la signification particulière diverge selon la source : la *quilisma*, l’*oriscus* et le *strophicus*. Ils s’obtiennent en préfixant la hauteur d’un `\quilisma`, `\oriscus` ou `\strophica`.

Il est virtuellement possible d’agglutiner autant de notes que voulu, y compris en les mélangeant avec des `\pes`, `\flexa`, `\virga`, `\inclinatum`, et de borner le tout par `\[` et `\]` pour produire une seule ligature. C’est d’ailleurs de cette manière que nous avons procédé pour générer le tableau qui suit. La création de ligatures est donc sans limite.

Notez bien que l’utilisation de ces signes en musique suit un certain nombre de règles, et que LilyPond n’effectue aucun contrôle à ce niveau. Par exemple, une *quilisma* se trouve toujours être la note intermédiaire d’une ligature ascendante et tombe habituellement sur un demi ton ;

bien que cela soit tout à fait possible, mais parfaitement incorrect, rien ne vous empêche de créer une quilisma sur une seule note.

En plus des signes propres à la notation, le fichier ‘`gregorian.ly`’ contient la définition des commandes `\versus`, `\responsum`, `\ij`, `\iij`, `\IJ` et `\IIJ`, qui permettent par exemple d’indiquer dans les paroles des repères de section. Ces commandes font appel à des caractères unicode spécifiques qui ne seront reproduits que si vous utilisez une fonte qui en dispose.

Le tableau ci-dessous inventorie, bien que dans une certaine limite, les différents neumes contenus dans le second tome de l’Antiphonale Romanum (*Liber Hymnarius*) publié par l’abbaye de Solesmes en 1983. La première colonne énumère le nom des ligatures – forme normale en gras et forme liquescente en italique. La troisième colonne contient le code ayant permis de générer la ligature, se basant ici sur `sol`, `la`, `si`.

Neumes simples

Formes Normale et <i>Liquescente</i>	Rendu	Code LilyPond
Punctum	■	<code>\[b \]</code>
	◻	<code>\[\cavum b \]</code>
	■	<code>\[\linea b \]</code>
<i>Punctum Auctum Ascendens</i>	┘	<code>\[\auctum \ascendens b \]</code>
<i>Punctum Auctum Descendens</i>	└	<code>\[\auctum \descendens b \]</code>
Punctum inclinatum	◆	<code>\[\inclinatum b \]</code>
<i>Punctum Inclinatum Auctum</i>	◆	<code>\[\inclinatum \auctum b \]</code>

Punctum Inclinatum Parvum

\[\inclinatum \deminutum b \]

•

Virga

┐

Ligatures sur deux notes**Clivis vel Flexa**

\[b \flexa g \]

┌┐

Clivis Aucta Descendens\[b \flexa \auctum \descendens
g \]

┌┐

Clivis Aucta Ascendens\[b \flexa \auctum \ascendens
g \]

┐┌

Cephalicus

\[b \flexa \deminutum g \]

┐

Podatus/Pes

\[g \pes b \]

┐┐

Pes Auctus Descendens\[g \pes \auctum \descendens b
\]

┐┐

Pes Auctus Ascendens\[g \pes \auctum \ascendens b
\]

┐┐

Epiphonus

\[g \pes \deminutum b \]

*Pes Initio Debilis*

\[\deminutum g \pes b \]

*Pes Auctus Descendens Initio Debilis*\[\deminutum g \pes \auctum
\descendens b \]**Ligatures sur plusieurs notes****Torculus**

\[a \pes b \flexa g \]

*Torculus Auctus Descendens*\[a \pes b \flexa \auctum
\descendens g \]*Torculus Deminutus*\[a \pes b \flexa \deminutum g
\]*Torculus Initio Debilis*\[\deminutum a \pes b \flexa g
\]*Torculus Auctus Descendens Initio
Debilis*\[\deminutum a \pes b \flexa
\auctum \descendens g \]*Torculus Deminutus Initio Debilis*\[\deminutum a \pes b \flexa
\deminutum g \]

Porrectus $\backslash[a \backslash flexa g \backslash pes b \backslash]$ *Porrectus Auctus Descendens* $\backslash[a \backslash flexa g \backslash pes \backslash auctum \backslash descendens b \backslash]$ *Porrectus Deminutus* $\backslash[a \backslash flexa g \backslash pes \backslash deminutum b \backslash]$ **Climacus** $\backslash[\backslash virga b \backslash inclinatum a \backslash inclinatum g \backslash]$ *Climacus Auctus* $\backslash[\backslash virga b \backslash inclinatum a \backslash inclinatum \backslash auctum g \backslash]$ *Climacus Deminutus* $\backslash[\backslash virga b \backslash inclinatum a \backslash inclinatum \backslash deminutum g \backslash]$ **Scandicus** $\backslash[g \backslash pes a \backslash virga b \backslash]$ *Scandicus Auctus Descendens* $\backslash[g \backslash pes a \backslash pes \backslash auctum \backslash descendens b \backslash]$ *Scandicus Deminutus* $\backslash[g \backslash pes a \backslash pes \backslash deminutum b \backslash]$ **Signes spéciaux**

Quilisma

\[g \pes \quilisma a \pes b \]

*Quilisma Pes Auctus Descendens*\[\quilisma g \pes \auctum
\descendens b \]**Oriscus**

\[\oriscus b \]

*Pes Quassus*

\[\oriscus g \pes \virga b \]

*Pes Quassus Auctus Descendens*\[\oriscus g \pes \auctum
\descendens b \]**Salicus**

\[g \oriscus a \pes \virga b \]

*Salicus Auctus Descendens*\[g \oriscus a \pes \auctum
\descendens b \]**(Apo)stroph**

\[\stroph a b \]

*Stroph Aucta*

\[\stroph \auctum b \]

**Bistroph**

\[\stroph a b \stroph a b \]



Tristropha

```
\[ \stroph a b \stroph a b
\stroph a b \]
```

Trigonus

```
\[ \stroph a b \stroph a b
\stroph a a \]
```

Commandes prédéfinies

LilyPond dispose des préfixes suivants : `\virga`, `\stroph a`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.

Les préfixes de note peuvent s'agglutiner, modulo quelques restrictions. Par exemple, on peut appliquer un `\descendens` ou un `\ascendens` à une note, mais pas les deux simultanément à une même note.

Deux notes adjacentes peuvent être reliées grâce aux commandes `\pes` ou `\flexa` pour marquer une ligne mélodique respectivement ascendante ou descendante.

Utilisez la fonction musicale unaire `\augmentum` pour ajouter des points d'augmentum.

Voir aussi

Glossaire musicologique : [Section "ligature" dans *Glossaire*](#).

Manuel de notation : [\[Ligatures mensurales\]](#), page 418, [\[Ligatures\]](#), page 410.

Problèmes connus et avertissements

Lorsqu'un `\augmentum` apparaît dans une ligature en fin de portée, son placement vertical peut être erroné. Pour y remédier, ajoutez un silence invisible, `s8` par exemple, comme dernière note de cette portée.

L'`\augmentum` devrait être implémenté en tant que préfixe plutôt qu'en tant que fonction unaire, afin qu'`\augmentum` puisse s'intégrer avec d'autres préfixes dans n'importe quel ordre.

2.9.4 Typographie de notation kiévienne**Contextes de notation kiévienne**

Tout comme pour les notations grégorienne et mensurale, les contextes prédéfinis `KievanVoice` et `KievanStaff` permettent de générer une partition en notation carrée. Ces contextes initialisent les propriétés de tous les autres contextes et objets graphiques à des valeurs adéquates, de telle sorte que vous pouvez tout de suite vous lancer dans la saisie de votre chant, comme ci-dessous :

```
\score {
  <<
    \new KievanVoice = "melody" \transpose c c' {
      \cadenzaOn
        c4 c c c c2 b,\longa
        \bar "kievan"
      }
    \new Lyrics \lyricsto "melody" {
      -- -- -- -- .
    }
  >>
```


}



Господи помилуй.

Voir aussi

Glossaire musicologique : [Section “notation kiévienne” dans *Glossaire*](#).

Problèmes connus et avertissements

LilyPond prend en charge la notation kiévienne du style synodal, correspondant au corpus du Saint Synode russe des années 1910, récemment réédité par les éditions du patriarcat de Moscou. LilyPond ne prend pas en charge les formes plus anciennes et moins répandues de notation kiévienne que l’on trouvait en Galicie pour noter le plain-chant ruthène.

Clefs kiéviennes

La notation kiévienne n’utilise qu’une seule clef – la clef « Tse-fa-ut » – qui indique la position du do :

```
\clef "kievan-do"
\override NoteHead #'style = #'kievan
c
```



Voir aussi

Glossaire musicologique : [Section “notation kiévienne” dans *Glossaire*](#), [Section “clef” dans *Glossaire*](#).

Manuel de notation : [\(undefined\) \[Clef\]](#), page [\(undefined\)](#).

Têtes de note kiéviennes

La notation kiévienne requiert l’utilisation du style de tête de note approprié. Vous devrez affecter `kievan` à la propriété `style` de l’objet `NoteHead`.

En notation kiévienne, la note finale d’une pièce apparaît souvent sous la forme d’une `\longa`. L’indication d’un récitatif – plusieurs syllabes sont chantées sur une même hauteur – s’effectue à l’aide d’une `\breve`. Voici ce à quoi ressemblent les différentes notes kiéviennes :

```
\autoBeamOff
\cadenzaOn
\override NoteHead #'style = #'kievan
b'1 b'2 b'4 b'8 b'\breve b'\longa
```



Voir aussi

Glossaire musicologique : [Section “notation kiévienne” dans *Glossaire*](#), [Section “tête de note” dans *Glossaire*](#).

Manuel de notation : [Section A.9 \[Styles de tête de note\]](#), page 641.

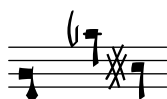
Problèmes connus et avertissements

LilyPond détermine automatiquement l'orientation des hampes. Il est cependant d'usage, en notation carrée, que les hampes des différentes notes d'un même mélisme aillent toutes dans le même sens ; il faudra donc en pareil cas définir manuellement la propriété `direction` de l'objet `Stem`.

Altérations kiéviennes

Le style d'altération propre à la notation kiévienne est accessible au travers de la propriété `glyph-name-alist` de l'objet `Accidental`. Le style `kievan` dispose d'un dièse et d'un bémol, tous deux différents du style par défaut ; il n'y a pas de bécarré en notation kiévienne. Bien que le dièse soit pas utilisé en notation synodale, on peut le trouver dans certains manuscrits plus anciens.

```
\clef "kievan-do"
\override NoteHead #'style = #'kievan
\override Accidental #'glyph-name-alist =
  #alteration-kievan-glyph-name-alist
bes' dis,
```



Voir aussi

Glossaire musicologique : [Section “notation kiévienne”](#) dans *Glossaire*, [Section “altération”](#) dans *Glossaire*.

Manuel de notation : [\[Altérations\]](#), page 5, [\[Altérations accidentelles automatiques\]](#), page 23, [Section A.8 \[La fonte Feta\]](#), page 620

Barre de mesure kiévienne

Les pièces en notation kiévienne sont habituellement terminées par une décoration qui fait office de barre finale. Elle s'obtient à l'aide d'un `\bar "kievan"`.

```
\clef "kievan-do"
\override NoteHead #'style = #'kievan
c \bar "kievan"
```



Voir aussi

[\[Barres de mesure\]](#), page 90, [Section A.8 \[La fonte Feta\]](#), page 620

2.9.5 Réédition de musique ancienne

Travailler sur de la musique ancienne requiert bien souvent des tâches particulières et qui s'éloignent fortement de la notation moderne pour laquelle LilyPond est conçu. Nous allons aborder, au fil des paragraphes qui suivent, un certain nombre de cas particuliers et vous proposer des suggestions, voire des solutions aux problèmes que vous ne manquerez pas de rencontrer. Ceci inclut entre autres :

- comment réaliser un incipit, autrement dit un court extrait montrant ce à quoi ressemblait l'original, en introduction à la transcription d'une œuvre médiévale ;

- comment obtenir une présentation *Mensurstriche* comme on peut le voir dans nombre de transcriptions de musique polyphonique ;
- comment transcrire du grégorien en notation moderne ;
- comment obtenir à la fois une reproduction en notation ancienne et une édition en notation moderne à partir d'une même source.

Des incipits

En cours de rédaction

Voir aussi

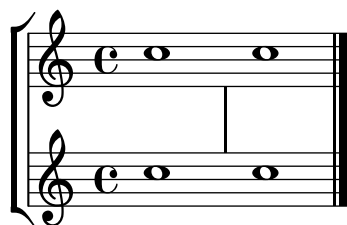
Morceaux choisis: [Section “Notations anciennes” dans *Morceaux choisis*](#).

Mise en forme de la musique mensurale

Mensurstriche, pour « lignes de mensuration », est le terme consacré lorsque les barres de mesure apparaissent uniquement entre les portées d'un système. Cette présentation permet de préserver l'aspect rythmique de l'original – par exemple sans couper une syncope par l'apparition d'une barre – tout en procurant l'aide que peuvent constituer les barres de mesure.

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat avec un `StaffGroup` plutôt qu'en utilisant un `ChoirStaff`, il faudra rendre « transparentes » les portions de barre qui recouvrent les portées.

```
global = {
  \override Staff.BarLine #'transparent = ##t
  s1 s
  % the final bar line is not interrupted
  \revert Staff.BarLine #'transparent
  \bar "|."
}
\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```



Transcription de chant grégorien

Une transcription d'un chant grégorien en notation moderne s'obtient grâce à quelques simples artifices.

Hampes. La gravure des hampes s'annule en supprimant le graveur `Stem_engraver` du contexte de voix :

```
\layout {
  ...
  \context {
```

```

\Voice
  \remove "Stem_engraver"
}
}

```

Certaines transcriptions laissent néanmoins apparaître occasionnellement des hampes, notamment pour indiquer la transition entre un récitatif monodique et une phrase mélodique. Il suffit en pareil cas d'utiliser plutôt `\override Stem #'transparent = ##t` ou `\override Stem #'length = #0` puis, en cas de besoin, recourir à une clause `\once \override Stem #'transparent = ##f` comme dans l'exemple ci-dessous. N'oubliez pas que les crochets aussi doivent disparaître, grâce à un `\override Flag #'transparent = ##t`.

Temps. En matière de chant non mesuré, plusieurs alternatives s'offrent à vous.

La suppression du `Time_signature_engraver` du contexte `Staff` ne produit aucun effet négatif. Une alternative serait de rendre la métrique transparente, ce qui par contre préservera l'espace qu'elle occupe.

Dans de nombreux cas, une clause `\set Score.timing = ##f` donne de bons résultats. On pourrait aussi utiliser `\CadenzaOn` et `\CadenzaOff`.

Rien de plus radical que de supprimer du contexte `Staff` le `Bar_engraver` pour ne pas voir de barre de mesure. Là aussi, une clause `\override BarLine #'transparent = ##t` vous permettra d'en afficher une au besoin.

Dans de nombreuses transcriptions, le récitatif fait apparaître une brève au lieu de la répétition d'une même note. Le texte psalmodié se présente alors sous la forme d'une unique syllabe alignée à gauche :

```

\include "gregorian.ly"
chant = \relative c' {
  \clef "G_8"
  c\breve c4 b4 a c2 c4 \divisioMaior
  c\breve c4 c f, f \finalis
}

verba = \lyricmode {
  \once \override LyricText #'self-alignment-X = #-1
  "Noctem quietam et" fi -- nem per -- fec -- tum
  \once \override LyricText #'self-alignment-X = #-1
  "concedat nobis Dominus" om -- ni -- po -- tens.
}

\score {
  \new Staff <<
  \new Voice = "melody" \chant
  \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \override Stem #'transparent = ##t
      \override Flag #'transparent = ##t
    }
  }
}

```

}



tens.

Ceci fonctionne bien tant que le texte ne risque pas de déborder de la ligne. Si tel était le cas, on pourrait plutôt ajouter des notes masquées, ici en jouant aussi sur la visibilité des hampes :

```
\include "gregorian.ly"
chant = \relative c' {
  \clef "G_8"
  \set Score.timing = ##f
  c\breve \override NoteHead #'transparent = ##t c c c c c
  \revert NoteHead #'transparent
  \override Stem #'transparent = ##f \stemUp c4 b4 a
  \override Stem #'transparent = ##t
  \override Flag #'transparent = ##t c2 c4 \divisioMaior
  c\breve \override NoteHead #'transparent = ##t c c c c c c c
  \revert NoteHead #'transparent c4 c f, f \finalis
}

verba = \lyricmode {
  No -- ctem qui -- e -- tam et fi -- nem per -- fec -- tum
  con -- ce -- dat no -- bis Do -- mi -- nus om -- ni -- po -- tens.
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics \lyricsto "melody" \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \override BarLine #'transparent = ##t
      \override Stem #'transparent = ##t
      \override Flag #'transparent = ##t
    }
  }
}
```



Autre situation courante, la transcription de chant neumatique contenant des mélismes, autrement dit, une psalmodie dans laquelle le nombre de syllabes varie selon les notes. Vous pourriez alors avoir envie d'indiquer clairement le découpage des groupes de syllabes ainsi que les subdivisions d'un mélisme. Le moyen pour y parvenir consiste à utiliser une métrique fixe, mettons `\time 1/4`, et de faire en sorte que chaque syllabe ou groupe de notes tienne dans une mesure, à l'aide de triolets ou de durées inférieures. Tant que les barres de mesure et autres éléments rythmiques restent transparents, et que l'espacement en regard des barres est accru, la représentation en notation moderne devrait être tout à fait satisfaisante.

Pour une répartition plus homogène de syllabes de longueur différente – telles que « -ri » et « -rum » – selon les groupes de note, une solution consiste à figer la propriété '`X-extent`' de l'objet `LyricText`. Ceci s'avère moins fastidieux que d'ajouter des syllabes sous forme de *markup*. Des ajustements supplémentaires peuvent se réaliser avec des « notes silencieuses » (`s`).

```
spiritus = \relative c' {
  \time 1/4
  \override Lyrics.LyricText #'X-extent = #'(0 . 3)
  d4 \times 2/3 { f8 a g } g a a4 g f8 e
  d4 f8 g g8 d f g a g f4 g8 a a4 s
  \times 2/3 { g8 f d } e f g a g4
}

spirLyr = \lyricmode {
  Spi -- ri -- _ _ tus _ Do -- mi -- ni _ re -- ple -- _ vit _
  or -- _ bem _ ter -- ra -- _ rum, al -- _ _ le -- _ lu
  -- _ ia.
}

\score {
  \new Staff <<
    \new Voice = "chant" \spiritus
    \new Lyrics = "one" \lyricsto "chant" \spirLyr
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \override BarLine #'X-extent = #'(-1 . 1)
      \override Stem #'transparent = ##t
      \override Flag #'transparent = ##t
      \override Beam #'transparent = ##t
      \override BarLine #'transparent = ##t
      \override TupletNumber #'transparent = ##t
    }
  }
}
```





Éditions ancienne et moderne à partir d'une même source

En cours de rédaction

Notation éditoriale

En cours de rédaction

2.10 Musiques du monde

Ce chapitre a pour objet la notation des musiques traditionnelles autres qu'occidentales.

2.10.1 Noms des notes et altérations non-occidentaux

Nous allons voir ici comment saisir et imprimer des partitions dans d'autres formes que la musique occidentale, que les anglophones appellent aussi *Common practice period*.

Extension des systèmes de notation et d'accordage

Les formes de notation propres à la musique classique traditionnelle sont employées dans toutes sortes de musique autres que le « classique ». Nous en avons déjà parlé dans le chapitre [Section 1.1.1 \[Écriture des hauteurs de note\]](#), [page 1](#), et plus particulièrement à la rubrique [\[Nom des notes dans d'autres langues\]](#), [page 7](#).

De nombreuses musiques autres qu'occidentales – et même certaines formes de musique traditionnelle occidentales – ont cependant recours à des systèmes de notation alternatifs ou étendus, qui ne s'intègrent pas forcément dans notre système standard.

Dans certains cas où la notation standard est utilisée, ces différences de hauteur seront implicites. Par exemple, la musique arabe est reproduite en notation standard et utilise des quarts de ton, l'altération réelle dépendant du contexte. Elle utilise traditionnellement la dénomination italienne, étendue dans le fichier 'arabic.ly' par un certain nombre de macros – voir [Section 2.10.2 \[Musique arabe\]](#), [page 438](#) pour plus de détails.

D'autres, par contre, font appel à une notation étendue, voire toute particulière. La *musique classique turque*, ou musique ottomane, utilise des formes mélodiques appelées *makamlar*, dans laquelle les tons sont divisés en neuf intervalles. Du point de vue actuel des pratiques de notation, il est possible d'utiliser les notes occidentales (do, ré, mi. . .) auxquelles on ajoutera l'altération spécifique à la musique turque. Ces différentes altérations sont définies dans le fichier 'makam.ly'. Pour de plus amples informations, reportez-vous à la rubrique [Section 2.10.3 \[Musique classique turque\]](#), [page 443](#).

Pour savoir où se trouvent les fichiers 'arabic.ly' et 'makam.ly' sur votre système, reportez vous au chapitre [Section "Autres sources de documentation"](#) dans *Manuel d'initiation*.

Morceaux choisis

Exemple de musique « Makam »

Le « makam » est une forme de mélodie turque qui utilise des altérations d'un neuvième de ton. Consultez le fichier d'initialisation 'makam.ly' pour plus de détails sur les hauteurs et altérations utilisées (voir le chapitre 4.6.3 - Autres sources d'information du manuel d'initiation pour le localiser).

```
% Initialize makam settings
```

```
\include "makam.ly"
```

```
\relative c' {
```

```
  \set Staff.keySignature = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
```


Noms des notes en arabe

Les noms de note les plus traditionnels en arabe peuvent être très longs et ne conviennent pas à l'écriture de la musique, aussi ne sont-ils pas utilisés. Les noms de note anglais ne sont pas très courants dans l'éducation musicale en arabe, c'est pourquoi on utilise plus volontiers les noms italiens (do, re, mi, fa, sol, la, si). On peut également utiliser des altérations, comme cela est expliqué dans [Nom des notes dans d'autres langues], page 7.

Par exemple, voici comment on peut écrire la gamme arabe *rast* :

```
\include "arabic.ly"
\relative do' {
  do re misb fa sol la sisb do sisb la sol fa misb re do
}
```



Le symbole indiquant un demi-bémol ne correspond pas au symbole utilisé dans la notation arabe. Si le symbole particulier du demi-bémol arabe doit absolument être utilisé, il est possible de s'en approcher en faisant précéder la note par la commande `\dwn` définie dans le fichier 'arabic.ly'. Cette méthode ne peut toutefois pas être utilisée pour modifier l'aspect du demi-bémol dans l'armure.

```
\include "arabic.ly"
\relative do' {
  \set Staff.extraNatural = ##f
  dod dob dosd \dwn dob dobsb dodsd do do
}
```



Voir aussi

Manuel de notation : [Section 2.10.1 \[Noms des notes et altérations non-occidentaux\]](#), page 437, [\[Nom des notes dans d'autres langues\]](#), page 7

Morceaux choisis : [Section "Musiques du monde"](#) dans *Morceaux choisis*.

Armures arabes

Outre les armures mineures et majeures, les armures suivantes sont définies dans la fichier 'arabic.ly' : *bayati*, *rast*, *sikah*, *iraq* et *kurd*. Ces armatures définissent un petit nombre de groupes de maqams plutôt que le grand nombre de maqams habituellement utilisés.

En général, un maqam utilise l'armure de son groupe ou d'un groupe voisin et diverses altérations accidentelles sont indiquées tout au long de la musique.

Par exemple, pour indiquer l'armure d'une pièce en maqam muhayer :

```
\key re \bayati
```

Ici, *re* est le nom de la tonalité par défaut de la base maqam dans le groupe.

Alors que l'armure correspond à un groupe, il est courant que le titre mentionne un maqam en particulier. Ainsi, dans cet exemple, le titre devrait faire apparaître le nom du maqam muhayer.

D'autres maqams du même groupe bayati, comme l'explique le tableau ci-dessous (bayati, hussaini, saba, et ushaq), peuvent être indiqués de la même manière. Ils sont autant de variations

du maqam de base, le plus courant, du groupe (en l'occurrence, bayati). En général, c'est dans les tétracordes supérieurs que ces modes apparentés diffèrent, ou dans certains détails de disposition qui ne changent pas fondamentalement leur nature.

Certains maqams ne sont qu'une modulation de leur maqam de base. Ainsi, dans ce même groupe de bayati, du maqam Nawa, dont la modulation est indiquée entre parenthèses dans le tableau. Les maqams arabes n'admettent que des modulations limitées, en raison de la nature des instruments de musique arabes. Le Nawa peut être indiqué comme suit :

```
\key sol \bayati
```

En musique arabe, le terme utilisé pour désigner un groupe maqam, tel que bayati, est également lui-même un maqam, généralement le plus important dans le groupe ; on peut le considérer comme un maqam de base.

Voici une suggestion de groupement qui relie les maqams les plus courants à leur armure :

groupe maqam	Armure	Tonique	Autres maqams dans le groupe (tonique)
ajam	major	sib	jaharka (fa)
bayati	bayati	re	hussaini, muhayer, saba, ushaq, nawa (sol)
hijaz	kurd	re	shahnaz, shad arban (sol), hijazkar (do)
irakien	iraq	sisb	-
kurde	kurd	re	kurde hijazkar (do)
nahawand	minor	do	busalik (re), farah faza (sol)
nakriz	mineur	do	nawa athar, hisar (re)
rast	rast	do	mahur, yakah (sol)
sikah	sikah	misb	huzam

Morceaux choisis

Armures inhabituelles

La commande `\key` détermine la propriété `keySignature` d'un contexte `Staff`.

Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété. Il s'agit en l'occurrence de définir une liste :

```
\set Staff.keySignature = #`(((octave . pas) . altération) ((octave . pas) . altération) ...)
```

dans laquelle, et pour chaque élément, `octave` spécifie l'octave (0 pour celle allant du do médium au si supérieur), `pas` la note dans cette octave (0 pour do et 6 pour si), et `altération` sera `,SHARP`, `,FLAT`, `,DOUBLE-SHARP` etc. (attention à la virgule en préfixe).

Une formulation abrégée – `(pas . altération)` – signifie que l'altération de l'élément en question sera valide quel que soit l'octave.

Voici, par exemple, comment générer une gamme par ton :

```
\relative c' {
  \set Staff.keySignature = #`(((0 . 6) . ,FLAT)
                                ((0 . 5) . ,FLAT)
                                ((0 . 3) . ,SHARP))

  c4 d e fis
  aes4 bes c2
}
```



Exemple de musique arabe

Voici un modèle qui utilise également le début d'un Semai turc courant dans l'éducation musicale arabe, pour illustrer quelques unes des particularités de la notation musicale arabe, comme des intervalles intermédiaires et des modes inhabituels traités dans ce chapitre.

```
\include "arabic.ly"
\score {
  \relative re' {
    \set Staff.extraNatural = ##f
    \set Staff.autoBeaming = ##f
    \key re \bayati
    \time 10/8

    re4 re'8 re16 [misb re do] sisb [la sisb do] re4 r8
    re16 [misb do re] sisb [do] la [sisb sol8] la [sisb] do [re] misb
    fa4 fa16 [misb] misb8. [re16] re8 [misb] re [do] sisb
    do4 sisb8 misb16 [re do sisb] la [do sisb la] la4 r8
  }
  \header {
    title = "Semai Muhayer"
    composer = "Jamil Bek"
  }
}
```



Voir aussi

Morceaux choisis : *Section "Musiques du mondes" dans Morceaux choisis.*

Lectures complémentaires pour la musique arabe

1. *La musique des Arabes* par Habib Hassan Touma [Amadeus Press, 1996], contient une étude des maqams et leur méthode de classification.

Il existe également de nombreux sites web qui expliquent les maqams, dont quelques uns s'accompagnent d'exemples audio :

- <http://www.maqamworld.com/>
- <http://www.turath.org/>

Si tout le monde s'accorde à apparenter les maqams d'après leur tétracorde inférieur, parfois transposé, les méthodes de classification varient dans certains détails.

2. Les sources ne sont pas entièrement cohérentes (parfois dans un même texte) quant à la manière d'indiquer l'armure de certains maqams. Il est courant, cependant, d'utiliser une armure par groupe plutôt qu'une armure différente pour chaque makam.

Des méthodes de luth arabe, l'*Oud*, par les auteurs suivants, contiennent des exemples de compositions principalement turques et arabes,

- Charbel Rouhana
- George Farah
- Ibrahim Ali Darwish Al-masri

2.10.3 Musique classique turque

Ce chapitre met en évidence des questions propres à la notation de la musique classique turque.

Références pour la musique classique turque

La musique classique turque s'est développée dans l'Empire Ottoman à peu près à la même période que la musique classique en Europe, et a continué jusqu'au XXe et XXIe siècle comme une tradition vibrante et distincte avec sa propre théorie, ses propres formes et styles d'interprétation. Parmi ses caractéristiques remarquables, se trouve l'usage de micro-intervalles fondés sur des « commas » d'un neuvième de ton, dont sont dérivées les formes mélodiques *makam* (pluriel *makamlar*).

Quelques questions relatives à la musique classique turque sont traitées dans d'autres chapitres :

- Les noms de notes et altérations sont mentionnés dans [Section 2.10.1 \[Noms des notes et altérations non-occidentaux\]](#), page 437.

Noms de note en turc

La musique classique turque attribue traditionnellement un nom unique à chaque hauteur, et du fait de la division du ton en neuf parts, les makamlar emploient une échelle de hauteurs complètement différente des gammes et modes d'occident :

koma de 1/9 de ton entier, *eksik bakiye* (3/9), *bakiye* (4/9), *küçük mücenneb* (5/9), *büyük mücenneb* (8/9), *tanîni* (un ton entier) et *artık ikili* (12/9 ou 13/9 de ton).

D'un point de vue de notation moderne, il est pratique d'utiliser positions occidentales des notes sur la portée (do, ré, mi. . .) avec des altérations spéciales qui haussent ou baissent les notes par intervalles de 1/9, 4/9, 5/9 et 8/9 de ton. Ces altérations sont définies dans le fichier 'makam.ly'.

Vous trouverez, dans le tableau suivant, le nom de ces altérations, le suffixe à utiliser, ainsi que la fraction de ton entier à laquelle elles correspondent.

Nom d'altération	suffixe	altération
büyük mücenneb (dièse)	-bm	+8/9
küçük mücenneb (dièse)	-k	+5/9
bakiye (dièse)	-b	+4/9
koma (dièse)	-c	+1/9
koma (bémol)	-fc	-1/9
bakiye (bémol)	-fb	-4/9
küçük mücenneb (bémol)	-fk	-5/9
büyük mücenneb (bémol)	-fbm	-8/9

Pour plus d'information sur les formes de notation non-occidentales, reportez-vous au chapitre [Section 2.10.1 \[Noms des notes et altérations non-occidentaux\]](#), page 437.

Voir aussi

Glossaire musicologique : Section “makam” dans *Glossaire*, Section “makamlar” dans *Glossaire*.

Manuel de notation : Section 2.10.1 [Noms des notes et altérations non-occidentaux], page 437.

3 Généralités en matière d'entrée et sortie

Nous n'allons pas, dans ce chapitre, parler directement de notation, mais plutôt du contenu des fichiers source et du résultat produit par LilyPond.

3.1 Agencement du code

LilyPond traite des fichiers textuels. Ces fichiers portent par convention une extension `.ly`.

3.1.1 Structure d'une partition

Un bloc `\score` contient obligatoirement une seule expression musicale délimitée par des accolades :

```
\score {
...
}
```

Note : Il ne doit y avoir qu'une seule expression musicale globale dans un bloc `\score`, et elle doit être bornée par une paire d'accolades.

Cette unique expression musicale peut être de n'importe quelle taille et contenir d'autres expressions musicales aussi complexes soient elles. Voici quelques exemples d'expression musicale :

```
{ c'4 c' c' c' }
```

```
{
```

```
  { c'4 c' c' c' }
```

```
  { d'4 d' d' d' }
```

```
}
```

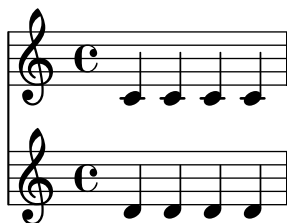


```
<<
```

```
\new Staff { c'4 c' c' c' }
```

```
\new Staff { d'4 d' d' d' }
```

```
>>
```



```
{
```

```
\new GrandStaff <<
```

```
  \new StaffGroup <<
```

```
    \new Staff { \flute }
```

```
    \new Staff { \hautbois }
```

```
  >>
```

```
\new StaffGroup <<
```

```
  \new Staff { \violonI }
```

```
  \new Staff { \violonII }
```

```
>>
>>
}
```

Les commentaires constituent l'une des rares exceptions à cette règle immuable – voir [Section 3.1.5 \[Structure de fichier\]](#), page 449 pour les autres. Qu'il s'agisse d'une seule ligne ou de tout un bloc – délimité par `%{ .. %}` – un commentaire peut se placer n'importe où dans le fichier source, aussi bien à l'intérieur qu'à l'extérieur du bloc `\score`, ou encore à l'intérieur ou à l'extérieur de l'expression musicale contenue dans un bloc `\score`.

Lorsqu'un fichier ne comprend qu'un bloc `\score`, celui-ci est implicitement inclus dans un bloc `\book`. Le bloc `\book` d'un fichier source permet la production d'au moins un fichier dont le nom sera, par défaut, déduit du fichier source : le traitement de `'fandangopourelephants.ly'` produira donc `'fandangopourelephants.pdf'`.

Pour de plus amples informations à propos du bloc `\book`, lisez [Section 3.1.2 \[Plusieurs partitions dans un même ouvrage\]](#), page 446, [Section 3.1.3 \[Plusieurs éditions pour une même source\]](#), page 447 et [Section 3.1.5 \[Structure de fichier\]](#), page 449.

Voir aussi

Manuel d'initiation : [Section “La partition est une \(unique\) expression musicale composée” dans *Manuel d'initiation*](#), [Section “Les expressions musicales en clair” dans *Manuel d'initiation*](#), [Section “Travail sur les fichiers d'entrée” dans *Manuel d'initiation*](#).

3.1.2 Plusieurs partitions dans un même ouvrage

Un ouvrage peut se composer de plusieurs morceaux et de texte. C'est le cas des cahiers d'exercices ou d'une partie d'orchestre avec ses différents mouvements. Chaque mouvement fait l'objet d'un bloc `\score`,

```
\score {
  ..musique..
}

et le texte est contenu dans un bloc \markup,
\markup {
  ..texte..
}
```

Les différents mouvements et textes qui apparaissent dans un même fichier `' .ly'` ne composeront en principe qu'un seul fichier résultant.

```
\score {
  ..
}
\markup {
  ..
}
\score {
  ..
}
```

Attention cependant si vous travaillez avec `lilypond-book` : il vous faudra explicitement mentionner le bloc `\book`, en l'absence de quoi seul le premier `\score` ou `\markup` apparaîtra après traitement.

L'entête de chaque pièce peut se placer au sein du bloc `\score` ; le contenu du champ `piece` viendra s'imprimer avant chaque mouvement. De même, le titre de l'ouvrage peut se placer au sein du bloc `\book`. Dans le cas contraire, le contenu du bloc `\header` placé en début de fichier sera utilisé.


```

\header {
  title = "Huit miniatures"
  composer = "Igor Stravinsky"
}
\score {
  ...
  \header { piece = "Romance" }
}
\markup {
  ..texte du second couplet..
}
\markup {
  ..texte du troisième couplet..
}
\score {
  ...
  \header { piece = "Menuet" }
}

```

Plusieurs pièces seront regroupées dans un même « chapitre » à l'aide d'un bloc `\bookpart`. Ces différents « chapitres » sont séparés par un saut de page et peuvent comporter un titre à l'instar de l'ouvrage dès lors que vous y insérez un bloc `\header`.

```

\bookpart {
  \header {
    title = "Titre de l'ouvrage"
    subtitle = "Première partie"
  }
  \score { ... }
  ...
}
\bookpart {
  \header {
    subtitle = "Deuxième partie"
  }
  \score { ... }
  ...
}

```

3.1.3 Plusieurs éditions pour une même source

Dès lors que vous inscrivez plusieurs blocs `\book` dans un même fichier `‘.ly’`, chacun d'eux donnera lieu à un résultat indépendant. Lorsqu'aucun bloc `\book` n'est spécifié dans le fichier source, LilyPond considère que l'intégralité du fichier constitue un bloc `\book` unique, comme indiqué à la rubrique [Section 3.1.5 \[Structure de fichier\], page 449](#).

LilyPond fait en sorte, lorsque plusieurs fichiers sont produits à partir d'une même source, qu'aucun résultat d'un bloc `\book` n'écrase celui qui a été généré pour un bloc `\book` précédent.

Dans les faits, et si le nom du fichier produit est repris de sa source – comportement par défaut –, un suffixe lui sera ajouté pour chaque `\book`. Il s'agit en principe d'un pseudo numéro de version. Ainsi, le fichier `‘huitminiatures.ly’` qui contiendrait

```

\book {
  \score { ... }
  \layout { ... }
}

```

```
\book {
  \score { ... }
  \layout { ... }
}
```

```
\book {
  \score { ... }
  \layout { ... }
}
```

génèrera

- ‘huitminiatures.pdf’,
- ‘huitminiatures-1.pdf’ et
- ‘huitminiatures-2.pdf’.

3.1.4 Nom des fichiers de sortie

LilyPond vous permet de prendre le contrôle dans la dénomination des fichiers que vous voulez générer, quel que soit le moteur de rendu utilisé.

Nous avons vu dans la rubrique précédente que LilyPond évite les conflits de nom des fichiers qu’il génère à partir d’une même source. Vous pouvez même définir vous-même le suffixe qui sera appliqué à chacun des blocs `\book`. Ainsi, en reprenant l’exemple ci-avant, vous obtiendrez les fichiers ‘huitminiatures-Romance.pdf’, ‘huitminiatures-Menuet.pdf’ et ‘huitminiatures-Nocturne.pdf’ en ajoutant simplement une déclaration `\bookOutputSuffix` au sein de chaque bloc `\book`.

```
\book {
  \bookOutputSuffix "Romance"
  \score { ... }
  \layout { ... }
}
\book {
  \bookOutputSuffix "Menuet"
  \score { ... }
  \layout { ... }
}
\book {
  \bookOutputSuffix "Nocturne"
  \score { ... }
  \layout { ... }
}
```

La déclaration `\bookOutputName` vous permet de définir vous-même le nom du fichier généré pour un bloc `\book` :

```
\book {
  \bookOutputName "Romance"
  \score { ... }
  \layout { ... }
}
\book {
  \bookOutputName "Menuet"
  \score { ... }
  \layout { ... }
}
\book {
```

```

\bookOutputName "Nocturne"
\score { ... }
\layout { ... }
}

```

Le traitement de ce fichier produira :

- 'Romance.pdf',
- 'Menuet.pdf' et
- 'Nocturne.pdf'.

3.1.5 Structure de fichier

Un fichier .ly peut contenir un certain nombre d'expressions de haut niveau. Les expressions de haut niveau sont les suivantes :

- Une définition de sortie, comme `\paper`, `\midi` et `\layout`. Ces définitions, lorsqu'elles se trouvent à un niveau supérieur, s'appliqueront à l'intégralité de l'ouvrage. Si l'une de ces expressions apparaît à plusieurs reprises à un niveau supérieur, les différents contenus seront combinés, à ceci près qu'en cas de déclarations conflictuelles, la dernière aura préséance. Des informations complémentaires sont disponibles à la rubrique [Section 4.2.1 \[Le bloc \layout\]](#), [page 507](#).
- Une expression Scheme pure, telle que `#{set-default-paper-size "a7" 'landscape}` ou `#{ly:set-option 'point-and-click #f}`.
- Un bloc `\header`, dont le contenu sera valide pour tout le fichier. Il comporte en général les valeurs par défaut des champs de titrage, tels le titre ou l'auteur entre autres, communs à tous les blocs `\book` inclus dans le fichier – voir [\[Généralités en matière de titrages\]](#), [page 451](#).
- Un bloc `\score` pour la partition. Cette partition sera assemblée avec les autres partitions se trouvant au même niveau pour composer le `\book`. Vous pouvez modifier ce comportement à l'aide de la variable `toplevel-score-handler` placée en tête. Le gestionnaire par défaut est défini dans le fichier d'initialisation '`../scm/lily.scm`'.
- Un bloc `\book` permet de regrouper naturellement plusieurs mouvements – autrement dit plusieurs blocs `\score` – dans un même document. Lorsqu'il y a plusieurs `\score`, LilyPond génère un seul fichier dans lequel les mouvements sont mis les uns à la suite des autres, ce pour chacun des blocs `\book` rencontrés. La seule raison qui peut vous demander d'explicitement plusieurs blocs `\book` dans un fichier '.ly' est lorsque vous avez besoin de générer différents documents à partir d'une même source. La présence explicite d'un bloc `\book` est aussi nécessaire lorsque vous travaillez sur un document lilypond-book qui reprendrait plusieurs `\score` ou `\markup` dans un même extrait. Vous pouvez modifier ce comportement à l'aide de la variable `toplevel-book-handler` placée en tête. Le gestionnaire par défaut est défini dans le fichier d'initialisation '`../scm/lily.scm`'.
- Un bloc `\bookpart`. Un ouvrage peut se découper en plusieurs parties à l'aide de blocs `\bookpart`, aussi bien pour alléger le travail de l'algorithme de calcul des sauts de page, que si les réglages du bloc `\paper` diffèrent d'une partie à l'autre.
- Une expression musicale telle que


```
{ c'4 d' e'2 }
```

Ce bout de code sera placé dans un `\score` et intégré à l'ouvrage en même temps que tous les autres `\score` ou expressions musicales. En d'autres termes, un fichier qui ne contiendrait que cette simple expression musicale sera traduit en

```

\book {
  \score {
    \new Staff {

```

```

    \new Voice {
      { c'4 d' e'2 }
    }
  }
  \layout { }
}
\paper { }
\header { }
}

```

Vous pouvez modifier ce comportement à l'aide de la variable `toplevel-music-handler` placée en tête. Le gestionnaire par défaut est défini dans le fichier d'initialisation `'../scm/lily.scm'`.

- Du texte sous forme de *markup* comme les paroles d'un couplet

```

\markup {
  2. Le première ligne du deuxième couplet.
}

```

De tels *markups* seront imprimés là où ils apparaissent, avant, après ou entre les expressions musicales.

- Une variable, ou identificateur, telle que

```
toto = { c4 d e d }
```

Vous pourrez la réutiliser plus loin dans votre fichier en saisissant simplement `\toto`. Le nom des identificateurs ne doit être formés que de caractères alphabétiques – sans chiffre ni caractère souligné ou tiret.

Voici trois éléments que vous pouvez placer à un niveau supérieur :

```

\layout {
  % pas en pleine largeur
  ragged-right = ##t
}

```

```

\header {
  title = "Do-re-mi"
}

```

```
{ c'4 d' e2 }
```

Vous pouvez placer, n'importe où dans votre fichier, les instructions suivantes :

- `\version`
- `\include`
- `\sourcefilename`
- `\sourcefileline`
- Une ligne de commentaire, introduite par le signe `%`.
- Un bloc de commentaire, délimité par `%{ .. %}`.

Vous pouvez insérer des espaces dans votre fichier source afin de lui apporter une meilleure lisibilité. Les espaces superflus sont normalement ignorés. Notez cependant qu'il est des cas où l'espace est requis pour éviter tout risque d'erreur :

- Autour d'une accolade, qu'elle soit ouvrant ou fermante ;
- Après chaque commande ou variable, autrement dit tout élément qui commence par un `\` ;
- Après tout élément qui sera interprété comme une expression Scheme, autrement dit tout élément qui commence par un `#` ;

- Pour séparer les éléments d'une expression Scheme ;
- En mode parole – `lyricmode` – pour séparer les termes des commandes `\override` et `\set`. Précisons à ce sujet qu'en plus d'ajouter une espace avant et après l'intégralité de la commande, vous devrez encadrer d'espaces le point et le signe égal qu'elle peut contenir, comme dans `\override Score . LyricText #'font-size = #5`.

Voir aussi

Manuel d'initiation : [Section “Organisation des fichiers LilyPond” dans *Manuel d'initiation*](#).

Manuel de notation : [\[Généralités en matière de titrages\], page 451, Section 4.2.1 \[Le bloc `\layout`\], page 507](#).

3.2 Titres et entêtes

La plupart de la musique qui est éditée comporte un titre et le nom de son compositeur ; certains ouvrages dispensent beaucoup plus d'informations.

3.2.1 Création de titres et entête ou pied de page

Généralités en matière de titrages

Chaque bloc `\book` apparaissant dans un même fichier source résultera en un fichier indépendant, comme indiqué à la rubrique [Section 3.1.5 \[Structure de fichier\], page 449](#). Chacun de ces fichiers résultants comporte deux endroits où placer des titrages : les **titrages de partie** au début de chaque partie (*bookpart*) et les **titrages de morceau** avant chaque pièce (*score*). Tous deux peuvent comporter les mêmes champs bien que, par défaut, le titrage d'un morceau se limite à *piece* et *opus*.

Les blocs `\header` peuvent se placer à quatre endroits différents qui formeront une hiérarchie descendante :

- En tête du fichier source, avant même tout bloc `\book`, `\bookpart` ou `\score` ;
- Au sein d'un bloc `\book` et en dehors de tout bloc `\bookpart` ou `\score` qu'il contient ;
- Au sein d'un bloc `\bookpart` et en dehors de tout bloc `\score` qu'il contient ;
- Après l'expression musicale incluse dans un bloc `\score`.

La valeur des différents champs sera filtrée en respectant cette hiérarchie ; les valeurs persisteront à moins d'être écrasées par une autre valeur à un niveau inférieur. Ainsi :

- Le titre d'une partie découle des champs définis en tête du fichier source, modifiés par les champs définis au sein du bloc `\book` puis par ceux définis au sein du bloc `\bookpart`. Les valeurs qui en résulteront permettront d'imprimer les titrages de partie pour cette partie.
- Le titre d'un morceau découle des champs définis en tête du fichier source, modifiés par les champs définis au sein du bloc `\book` puis par ceux définis au sein du bloc `\bookpart`, et enfin par ceux définis au sein du bloc `\score`. Les valeurs qui en résulteront permettront d'imprimer les titrages de morceau pour ce morceau. Notez toutefois que, pour un morceau, seuls les champs *piece* et *opus* seront imprimés, à moins d'avoir valorisé à `#t` la variable `print-all-headers` dans la section `\paper`.

Note : N'oubliez pas que lorsqu'il est placé à l'intérieur d'un bloc `\score`, le bloc `\header` doit impérativement se trouver **à la suite** de l'expression musicale.

Nul n'est besoin de fournir un bloc `\header` à chacun des quatre niveaux ; on peut se passer aussi bien de l'un d'eux que de tous. Dans la même veine, un fichier source simpliste peut ne pas mentionner de bloc `\book` ou `\bookpart` qui seront alors créés implicitement.

Lorsque l'ouvrage ne comporte qu'un seul morceau, le bloc `\header` devrait prendre place en tête de fichier, de telle sorte que soit produit un titrage de partie qui met à disposition tous les champs de titrage.

Lorsque l'ouvrage comporte plusieurs morceaux, différents arrangements du bloc `\header` permettent d'obtenir différents styles de publication musicale. Par exemple, si la publication comprend plusieurs pièces du même compositeur, un bloc `\header` placé en tête de fichier définira le titre de l'ouvrage et le compositeur, que l'on complètera par un bloc `\header` dans chaque bloc `\score` pour définir les champs `piece` et `opus`, comme ici :

```
\header {
  title = "SUITE I."
  composer = "J. S. Bach."
}

\score {
  \new Staff \relative g, {
    \clef bass
    \key g \major
    \repeat unfold 2 { g16( d' b') a b d, b' d, } |
    \repeat unfold 2 { g,16( e' c') b c e, c' e, } |
  }
  \header {
    piece = "Prélude."
  }
}

\score {
  \new Staff \relative b {
    \clef bass
    \key g \major
    \partial 16 b16 |
    <g, d' b'~>4 b'16 a( g fis) g( d e fis) g( a b c) |
    d16( b g fis) g( e d c) b(c d e) fis( g a b) |
  }
  \header {
    piece = "Allemande."
  }
}
```

SUITE I.

J. S. Bach.

Prélude.



Allemande.



Des agencements plus élaborés sont aussi réalisables. Par exemple, les champs appartenant au titrage principal d'un ouvrage peuvent se reporter dans chaque bloc `\score`, certains étant modifiés voire supprimés manuellement :

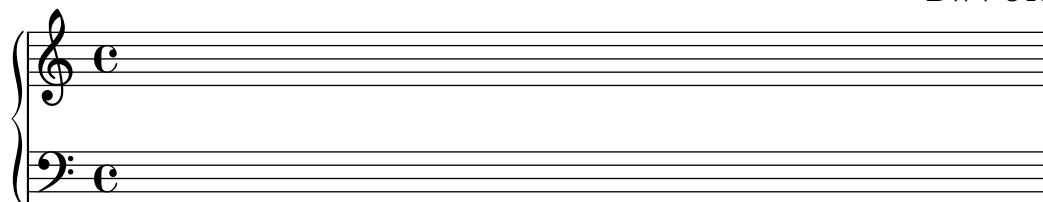
```
\book {
  \paper {
    print-all-headers = ##t
  }
  \header {
    title = "DAS WOHLTEMPERIRTE CLAVIER"
    subtitle = "TEIL I"
    % Pas de mention spéciale pour cet ouvrage
    tagline = ##f
  }
  \markup { \vspace #1 }
  \score {
    \new PianoStaff <<
      \new Staff { s1 }
      \new Staff { \clef "bass" s1 }
    >>
    \header {
      title = "PRAELUDIUM I"
      opus = "BWV 846"
      % Pas de sous-titre pour ce morceau
      subtitle = ##f
    }
  }
}
\score {
  \new PianoStaff <<
    \new Staff { s1 }
    \new Staff { \clef "bass" s1 }
  >>
  \header {
    title = "FUGA I"
    subsubtitle = "A 4 VOCI"
    opus = "BWV 846"
    % Pas de sous-titre pour ce morceau
    subtitle = ##f
  }
}
}
```

DAS WOHLTEMPERIRTE CLAVIER

TEIL I

PRAELUDIUM I

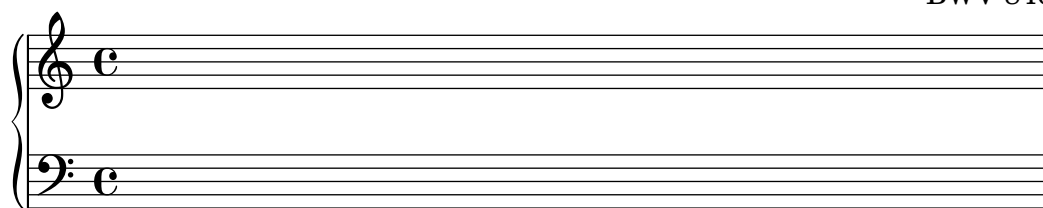
BWV 846



FUGA I

A 4 VOCI

BWV 846



Voir aussi

Manuel de notation : [Section 3.1.5 \[Structure de fichier\]](#), page 449, [\[Mise en forme par défaut des titrages subalternes\]](#), page 454, [\[Mise en forme personnalisée des titrages\]](#), page 459.

Mise en forme par défaut des titrages subalternes

Voici les différentes variables attachées au bloc `\header` :

```
\book {
  \header {
    % Les champs suivants sont centrés
    dedication = "Dédicace"
    title = "Titre"
    subtitle = "Sous-titre"
    subsubtitle = "Sous-sous-titre"
    % Les champs suivants sont répartis sur une même ligne, et
    % le champ "instrument" apparaîtra sur les pages suivantes
    instrument = \markup \with-color #green "Instrument"
    poet = "Librettiste"
    composer = "Compositeur"
    % Les champs suivants sont en opposition sur la même ligne
    meter = "Tempo"
    arrangeur = "Arrangeur"
    % Les champs suivants sont centrés en bas de page
    tagline = "« tagline » ou mention spéciale en pied de dernière page"
    copyright = "copyright en pied de première page"
  }
  \score {
```



```

{ s1 }
\header {
  % Les champs suivants sont en opposition sur la même ligne
  piece = "Pièce 1"
  opus = "Opus 1"
}
}
\score {
  { s1 }
  \header {
    % Les champs suivants sont en opposition sur la même ligne
    piece = "Pièce 2 sur la même page"
    opus = "Opus 2"
  }
}
\pageBreak
\score {
  { s1 }
  \header {
    % Les champs suivants sont en opposition sur la même ligne
    piece = "Pièce 3 sur une nouvelle page"
    opus = "Opus 3"
  }
}
}

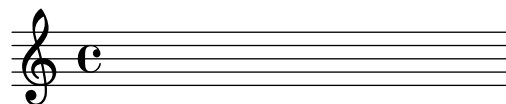
```

Dédicace

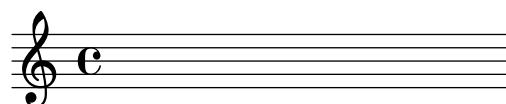
Titre**Sous-titre****Sous-sous-titre**Librettiste **Instrument** Compositeur

Tempo Arrangeur

Pièce 1 Opus 1

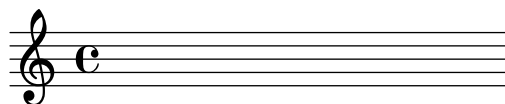


Pièce 2 sur la même page Opus 2



copyright en pied de première page

2 **Instrument**
 Pièce 3 sur une nouvelle page Opus 3



« tagline » ou mention spéciale en pied de dernière page

Quelques précisions :

- Le nom de l'instrument sera répété en tête de chaque page.
- Seuls seront imprimés les champs `piece` et `opus` inclus dans un bloc `\score` dès lors que la variable `print-all-headers` reste désactivée (valeur à `##f`).
- Les champs d'un bloc `\header` qui n'auront pas été alimentés seront remplacés par un `markup \null` de façon à ne pas gaspiller d'espace.
- Par défaut, `scoreTitleMarkup` place les champs `piece` et `opus` de part et d'autre sur une même ligne.

Les possibilités de modifier la mise en forme par défaut sont abordées à la rubrique [\[Mise en forme personnalisée des titrages\]](#), page 459.

La variable `breakbefore` activée dans un bloc `\header` situé dans un bloc `\score` force le saut de page avant le morceau contenu dans ce `\score`. Vous pourrez ainsi séparer le titre principal de la musique.

```
\book {
  \header {
    title = "This is my Title"
    subtitle = "This is my Subtitle"
    copyright = "This is the bottom of the first page"
  }
  \score {
    \repeat unfold 4 { e'' e'' e'' e'' }
    \header {
      piece = "This is the Music"
      breakbefore = ##t
    }
  }
}
```

This is my Title

This is my Subtitle

This is the bottom of the first page



Music engraving by LilyPond 2.16.2—www.lilypond.org

Voir aussi

Manuel d'initiation : [Section “Organisation des fichiers LilyPond”](#) dans *Manuel d'initiation*.

Manuel de notation : [\[Mise en forme personnalisée des titrages\]](#), page 459, [Section 3.1.5 \[Structure de fichier\]](#), page 449.

Fichiers d'initialisation : `'ly/titling-init.ly'`.

Mise en forme par défaut des entête et pied de page

Les entête et pied – *header* et *footer* – sont des lignes de textes qui apparaissent en haut et en bas de chaque page, indépendamment du texte de l'ouvrage. Ils sont contrôlés par les variables suivantes, attachées au bloc `\paper` :

- `oddHeaderMarkup` – entête de page impaire
- `evenHeaderMarkup` – entête de page paire
- `oddFooterMarkup` – pied de page impaire
- `evenFooterMarkup` – pied de page paire

Ces variables *markup* n'accèdent qu'au contenu des champs du bloc `\header` principal, celui qui s'appliquera à tous les blocs `\score` du document. Ces variables sont définies dans le fichier `'ly/titling-init.ly'`, et sont par défaut :

- les numéros sont placés en haut à gauche (si pair) ou à droite (si impair) de chaque page à compter de la deuxième ;
- le contenu du champ `instrument` est centré en haut de chaque page à compter de la deuxième ;
- le texte du `copyright` est centré au bas de la première page ;
- le `tagline` – mention spéciale – se place au bas de la dernière page, ou bien sous le `copyright` s'il n'y a qu'une seule page.



Music engraving by LilyPond 2.16.2—www.lilypond.org

La mention spéciale se modifie en alimentant le champ `tagline` au niveau du bloc `\header` principal.

```
\book {
  \header {
    tagline = "... music notation for Everyone"
  }
  \score {
    \relative c' {
      c4 d e f
    }
  }
}
```



... music notation for Everyone

Pour supprimer le `tagline`, il suffit de lui assigner la valeur `##f`.

3.2.2 Titrages personnalisés

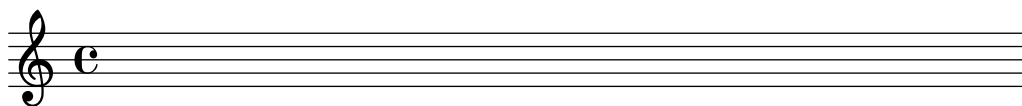
Mise en forme personnalisée des champs de titrage

Toutes les commandes de mise en forme d'un `\markup` permettent de personnaliser le texte des entête, pied de page et éléments de titrage contenus dans un bloc `\header`.

```
\score {
  { s1 }
  \header {
    piece = \markup { \fontsize #4 \bold "PRAELUDIUM I" }
    opus = \markup { \italic "(Excerpt)" }
  }
}
```

PRAELUDIUM I

(Excerpt)



Voir aussi

Manuel de notation : [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

Mise en forme personnalisée des titrages

L'utilisation de commandes `\markup` au sein d'un bloc `\header` permet de modifier aisément l'apparence du texte, mais n'influence en rien le positionnement précis des éléments de titrage. L'accès au positionnement des champs de titrage est géré par les deux variables suivantes, attachées au bloc `\paper` :

- `bookTitleMarkup`
- `scoreTitleMarkup`

Le positionnement des titres, avec les valeurs par défaut de ces variables `\markup`, est illustré à la rubrique [\[Mise en forme par défaut des titrages subalternes\]](#), page 454.

Voici les réglages par défaut de `scoreTitleMarkup`, tels que définis dans le fichier `'ly/titling-init.ly'` :

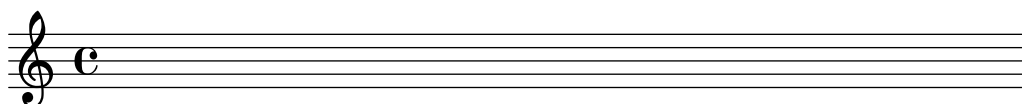
```
scoreTitleMarkup = \markup { \column {
  \on-the-fly #print-all-headers { \bookTitleMarkup \hspace #1 }
  \fill-line {
    \fromproperty #'header:piece
    \fromproperty #'header:opus
  }
}
```

Ceci aura donc pour effet de positionner les champs `piece` et `opus` sur la même ligne, en opposition :

```
\score {
  { s1 }
  \header {
    piece = "PRAELUDIUM I"
    opus = "BWV 846"
  }
}
```

PRAELUDIUM I

BWV 846



Voici comment redéfinir le `scoreTitleMarkup` de telle sorte que le champ `piece`, dont nous modifions la taille et la graisse, se place au centre de cette ligne :

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
      }
    }
  }
}
```

```

        \fromproperty #'header:opus
    }
}
\header { tagline = ##f }
\score {
  { s1 }
  \header {
    piece = "PRAELUDIUM I"
    opus = "BWV 846"
  }
}
}

```



Les champs normalement absents du `\header` d'un bloc `\score` seront toutefois imprimés dès lors que vous aurez activé l'instruction `print-all-headers` au sein du bloc `\paper`. Le principal inconvénient de cette fonction réside dans le fait que les champs dévolus au titrage des parties devront être supprimés dans chacun des blocs `\score` de votre fichier source – voir [\[Généralités en matière de titrages\]](#), page 451.

Afin d'éviter ce désagrément, ajoutez le champ que vous désirez voir apparaître à la définition de `scoreTitleMarkup`. Nous allons, dans l'exemple suivant, ajouter au `scoreTitleMarkup` le champ `composer`, normalement associé au `bookTitleMarkup` ; chaque `\score` pourra alors mentionner un compositeur différent.

```

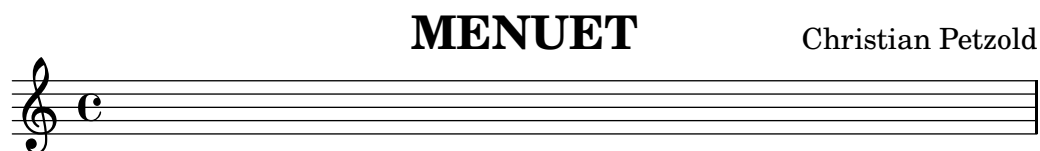
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:composer
      }
    }
  }
}
\header { tagline = ##f }
\score {
  { s1 }
  \header {
    piece = "MENUET"
    composer = "Christian Petzold"
  }
}
\score {
  { s1 }
  \header {
    piece = "RONDEAU"
  }
}

```

```

    composer = "François Couperin"
  }
}

```



Rien ne vous empêche de créer votre propre champ personnalisé, puis d'y faire référence dans la définition du *markup*.

```

\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \override #`(direction . ,UP) {
          \dir-column {
            \center-align \fontsize #-1 \bold
            \fromproperty #'header:mycustomtext %% User-defined field
            \center-align \fontsize #4 \bold
            \fromproperty #'header:piece
          }
        }
      }
      \fromproperty #'header:opus
    }
  }
}

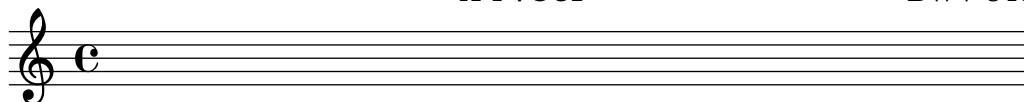
\header { tagline = ##f }
\score {
  { s1 }
  \header {
    piece = "FUGA I"
    mycustomtext = "A 4 VOICI" %% User-defined field
    opus = "BWV 846"
  }
}
}

```

FUGA I

A 4 VOCI

BWV 846



Voir aussi

Manuel de notation : [\[Généralités en matière de titrages\]](#), page 451.

Mise en forme personnalisée des entête et pied de page

L'utilisation de commandes `\markup` au sein d'un bloc `\header` permet de modifier aisément l'apparence du texte, mais n'influence en rien le positionnement précis des entête et pied de page. L'accès au positionnement des champs concernés est géré par les quatre variables suivantes, attachées au bloc `\paper` :

- `oddHeaderMarkup`
- `evenHeaderMarkup`
- `oddFooterMarkup`
- `evenFooterMarkup`

L'instruction `\on-the-fly` au sein d'un `\markup` permet d'ajouter, lorsque certaines conditions sont respectées, des éléments au texte des entête et pied de page définis dans le bloc `\paper`. En voici la syntaxe :

```
variable = \markup {
  ...
  \on-the-fly #procédure markup
  ...
}
```

La *procédure* est appelée à chaque fois que la commande `\markup` où elle apparaît est évaluée. La *procédure* effectuera un test de conformité particulier et interprètera, autrement dit imprimera l'argument *markup* si et seulement si cette condition est remplie.

LilyPond dispose d'ores et déjà d'un certain nombre de procédures :

Nom de la procédure	Condition testée
<code>print-page-number-check-first</code>	il faut imprimer ce numéro de page.
<code>create-page-number-stencil</code>	' <code>print-page-numbers</code> est vrai.
<code>print-all-headers</code>	' <code>print-all-headers</code> est vrai.
<code>first-page</code>	c'est la première page du <i>book</i> .
(<code>on-page</code> nombre)	numéro de page = nombre
<code>last-page</code>	c'est la dernière page du <i>book</i> .
<code>not-first-page</code>	ce n'est la première page du <i>book</i> .
<code>part-first-page</code>	c'est la première page de la partie.
<code>part-last-page</code>	c'est la dernière page de la partie.
<code>not-single-page</code>	cette partie fait plus d'une page.

L'exemple suivant illustre la manière de centrer son numéro au bas de chaque page. Il nous faut tout d'abord annuler les définitions de `oddHeaderMarkup` et `evenHeaderMarkup` à l'aide d'un *markup* `\null`. Nous redéfinissons ensuite `oddFooterMarkup` pour qu'il contienne le numéro de page, centré. Enfin, nous appliquons le même paramétrage au `\oddFooterMarkup`.

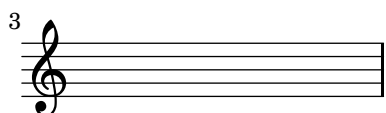
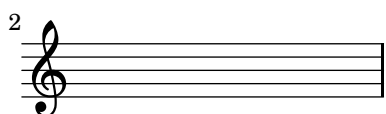
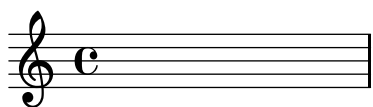
```
\book {
  \paper {
```



```

print-page-number = ##t
print-first-page-number = ##t
oddHeaderMarkup = \markup \null
evenHeaderMarkup = \markup \null
oddFooterMarkup = \markup {
  \fill-line {
    \on-the-fly #print-page-number-check-first
    \fromproperty #'page:page-number-string
  }
}
evenFooterMarkup = \oddFooterMarkup
}
\score {
  \new Staff { s1 \break s1 \break s1 }
}
}

```



1

Plusieurs conditions `\on-the-fly` mentionnées l'une à la suite de l'autre se cumulent. Ainsi, par exemple,

```

\on-the-fly #first-page
\on-the-fly #last-page
{ \markup ... \fromproperty #'header: ... }

```

teste si la sortie tient sur une page unique.

Voir aussi

Manuel de notation : [\[Généralités en matière de titrages\]](#), page 451, [\[Mise en forme par défaut des titrages subalternes\]](#), page 454.

fichiers d'initialisation : `'../ly/titling-init.ly'`.

3.2.3 Notes de bas de page

Les notes de bas de page sont utiles dans bien des situations. Dans tous les cas, un « appel de note » vient se placer en référence dans le texte ou la musique, et le « texte de la note » est reporté en bas de la page.

Selon qu'elle est référencée dans une expression musicale ou dans du texte indépendant, une note de bas de page sera créée suivant une procédure différente.

Notes de bas de page dans une expression musicale

Généralités sur l'annotation de musique

Il existe deux catégories d'annotation concernant une expression musicale :

Les annotations événementielles

se rattachent à des événements particuliers, comme une note individuelle ou au sein d'un accord, un élément d'interprétation (ligature, liaison, doigté, accent ou nuance) et des paroles.

Les annotations temporelles

se rapportent à un point particulier du déroulement d'un contexte musical. Certaines commandes, telles `\time` et `\clef`, ne reposent pas sur un événement pour la création de l'objet métrique ou clef. Il en va de même pour un accord : sa hampe ou ses crochets ne sont créés qu'à la fin d'un moment (plus exactement au travers de l'un des événements note qui le composent). Une annotation temporelle permet d'annoter ces objets de rendus sans se référer à un événement.

Examinons la forme développée d'une commande de note de bas de page, qu'elle soit événementielle ou temporelle :

`[position] \footnote [marque] décalage [nom-grob] annotation musique`

Ses arguments peuvent se définir ainsi :

- | | |
|-------------------|---|
| <i>position</i> | Lorsque la commande <code>\footnote</code> s'applique à un élément d'interprétation ou un événement rattaché, et uniquement dans ces cas, elle doit être précédée d'un indicateur de positionnement (<code>-</code> , <code>_</code> ou <code>^</code>) de façon à rattacher <i>musique</i> (avec sa marque) à la note ou au silence qui précède. |
| <i>marque</i> | Un <i>markup</i> ou une chaîne de caractères identifiant l'annotation tant au niveau de l'appel que de la note qui apparaîtra au bas de la page. L'absence de cet élément – ou une valeur de <code>\default</code> – incrémentera automatiquement le compteur. Ce compteur est réinitialisé à chaque page comportant une annotation. |
| <i>décalage</i> | Une paire de nombres – <code>'#(2 . 1)'</code> par exemple – spécifiant le décalage de la marque, en abscisse et en ordonnée, par rapport au point de référence. Des valeurs positives décalent vers la droite ou le haut, des valeurs négatives vers la gauche ou le bas ; des valeurs à zéro centrent la marque sur le point de référence. Le décalage s'exprime en espace de portée. |
| <i>nom-grob</i> | Le type d'objet à annoter – <code>'#Flag'</code> par exemple. Lorsque cet élément est spécifié, c'est l'objet en question qui servira de point de référence, même s'il trouve son origine non pas directement dans l'élément <i>musique</i> mais dans un objet en résultant. Son absence ou une valeur de <code>\default</code> ne permet d'annoter que des objets directement créés. |
| <i>annotation</i> | un <i>markup</i> ou une chaîne de caractères qui sera reporté au bas de la page. |
| <i>musique</i> | l'élément qui fait l'objet du commentaire, qu'il s'agisse d'un événement musical, de l'un des constituants d'un accord ou d'un événement rattaché. Bien qu'il ne puisse être omis, il peut être remplacé par <code>\default</code> , auquel cas l'annotation ne sera pas rattachée à une expression musicale particulière, mais plutôt à un instant précis. L'argument <i>nom-grob</i> est alors obligatoire pour sélectionner le type d'objet affecté – <code>'#TimeSignature'</code> par exemple. |

Notes de bas de page événementielles

Le moyen le plus simple d'annoter un événement est :

```
\footnote décalage annotation musicale
```

Ce type de note de bas de page s'attache à un objet de rendu généré par l'événement correspondant à *musique*.

```
\book {
  \header { tagline = ##f }
  \relative c' {
    a4_\footnote #'(0 . -1) "Une liaison arbitrairement en dessous" (
    b8^\footnote #'(1 . 0.5) "Une ligature manuelle forcée en haut" [
    b8 ]
    c4 )
    c-\footnote #'(1 . 1) "Tenuto" --
  }
}
```



-
- ¹Une liaison arbitrairement en dessous
²Une ligature manuelle forcée en haut
³Tenuto

Les appels de note peuvent être personnalisés, et le trait reliant l'objet à l'appel supprimé :

```
\book {
  \header { tagline = ##f }
  \relative c' {
    \footnote "*" #'(0.5 . -2) \markup { \italic "*" La première note" }
    a'4 b8
    \footnote \markup { \super "$" } #'(0.5 . 1)
    \markup { \super "$" \italic " La deuxième note" }
    e c4
    \once \override Score.FootnoteItem #'annotation-line = ##f
    b-\footnote \markup \tiny "+" #'(0.1 . 0.1)
    \markup { \super "+" \italic " Éditorial" } \p
  }
}
```



* *La première note*

\$ *La deuxième note*

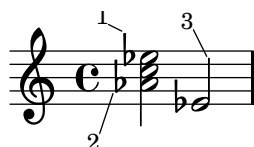
+ *Éditorial*

D'autres exemples de personnalisation des appels de note sont donnés à la rubrique [Notes de bas de page dans du texte indépendant], page 467.

L'annotation d'un accord dans sa globalité n'est pas réalisable de cette manière dans la mesure où un accord ne génère pas d'événement en dehors de ses constituants. Les notes qui le composent peuvent cependant faire l'objet d'une annotation.

Lorsque la note de bas de page se réfère à un objet de rendu résultant d'un événement – Accidental ou Stem découlent d'un NoteHead –, l'argument additionnel *nom-grob* est requis avant le texte de l'annotation :

```
\book {
  \header { tagline = ##f }
  \relative c'' {
    % footnotes may be added to chord constituents
    < \footnote #'(-1 . -3) #'Accidental "Un autre bémol" aes
      c
      \footnote #'(-1 . 0.5) #'Accidental "Un bémol" ees
    >2
    \footnote #'(-1 . 2) #'Stem "Une hampe" ees2
  }
}
```



¹Un bémol

²Un autre bémol

³Une hampe

Note : Lorsque plusieurs notes de bas de page se rapportent à un même empilement vertical, elles sont numérotées et apparaîtront selon l'ordre vertical des éléments présentés, autrement dit celui positionné le plus haut en premier, non dans leur ordre d'apparition dans le fichier source.

Notes de bas de page temporelles

Les objets de rendu tels que changement de clef ou d'armure tirent leur origine dans la modification d'une propriété plutôt que d'un véritable événement. D'autres, comme les barres ou numéros de mesure, dépendent directement de la temporisation. C'est la raison pour laquelle de tels objets doivent s'annoter en fonction de leur survenance au fil de la musique. Les notes

de bas de page temporelles sont la solution à privilégier lorsqu'il s'agit d'annoter les hampes ou ligature affectant des accords : bien qu'une telle fonctionnalité puisse s'appliquer à l'un des événements constituant l'accord, rien ne laisse présager lequel serait le plus adéquat.

Une note de bas de page temporelle s'introduit de la même manière qu'une note événementielle, à ceci près que la mention `\default` remplacera l'argument *musique* qui indique l'événement à annoter. En matière de note de bas de page temporelle, l'objet de rendu considéré devrait toujours être mentionné explicitement, de telle sorte qu'il n'y ait pas d'erreur quant au positionnement de l'appel de note.

```
\book {
  \header { tagline = ##f }
  \relative c'' {
    r1 |
    \footnote #'(-0.5 . -1) #'TimeSignature "Changement de métrique" \default
    \time 3/4
    \footnote #'(1 . -1) #'Stem "Hampe de l'accord" \default
    <c e g>4 q q
    \footnote #'(-0.5 . 1) #'BarLine "Barre de mesure" \default
    q q
    \footnote #'(0.5 . -1) #'KeySignature "Changement d'armure" \default
    \key c\minor
    q
  }
}
```



¹Changement de métrique

²Hampe de l'accord

³Barre de mesure

⁴Changement d'armure

Notes de bas de page dans du texte indépendant

De telles notes de bas de page affectent les *markup* extérieurs aux expressions musicales. Il n'est pas nécessaire en pareil cas d'indiquer un point de référence par un trait ; l'appel de note vient juste s'accoler au *markup* qui fait l'objet de l'annotation. Les appels de note peuvent être gérés automatiquement, auquel cas ils seront numériques, ou bien manuellement en fournissant un indicateur particulier.

Les notes de bas de page concernant du texte indépendant se gèrent différemment selon qu'elles sont automatiques ou manuelles.

Notes de bas de page automatiques dans du texte

La syntaxe consacrée dans le cas d'une gestion automatique des appels de note est :

```
\markup { ... \auto-footnote texte annotation ... }
```

Dont les éléments sont :

texte le *markup* ou la chaîne de caractères sur lequel porte l'annotation ;

annotation

un *markup* ou une chaîne de caractères constituant le texte de l'annotation qui sera reportée en bas de page.

Par exemple :

```
\book {
  \header { tagline = ##f }
  \markup {
    "A simple"
    \auto-footnote "tune" \italic " By me"
    "is shown below. It is a"
    \auto-footnote "recent" \italic " Aug 2012"
    "composition."
  }
  \relative c' {
    a'4 b8 e c4 d
  }
}
```

A simple tune¹ is shown below. It is a recent² composition.



¹ *By me*

² *Aug 2012*

Notes de bas de page personnalisées dans du texte

La syntaxe consacrée dans le cas d'une gestion personnalisée des appels de note est :

```
\markup { ... \footnote appel annotation ... }
```

Dont les éléments sont :

appel un *markup* ou une chaîne de caractères représentant l'appel de note affecté à ce point de référence. Notez bien que cette marque ne sera **pas** reproduite automatiquement avant le texte proprement dit de l'annotation.

annotation

un *markup* ou une chaîne de caractères constituant le texte de l'annotation qui sera reportée en bas de page, précédé de l'*appel*.

N'importe quel caractère simple tel que * ou + peut s'utiliser en tant qu'appel de note, comme nous l'avons vu à la rubrique [Notes de bas de page dans une expression musicale], page 464. D'autres caractères particuliers sont accessibles sous forme de raccourci – voir la rubrique [Équivalents ASCII], page 482 :

```

\book {
  \paper { #(include-special-characters) }
  \header { tagline = ##f }
  \markup {
    "A simple tune"
    \footnote "*" \italic "*" By me"
    "is shown below. It is a recent"
    \footnote \super &dagger; \concat {
      \super &dagger; \italic " Aug 2012"
    }
    "composition."
  }
  \relative c' {
    a'4 b8 e c4 d
  }
}

```

A simple tune * is shown below. It is a recent[†] composition.



* *By me*

† *Aug 2012*

Un appel de note peut aussi se libeller sous la forme d'un point de code unicode – voir la rubrique [\[Unicode\]](#), page 481 :

```

\book {
  \header { tagline = ##f }
  \markup {
    "A simple tune"
    \footnote \super \char##x00a7 \concat {
      \super \char##x00a7 \italic " By me"
    }
    "is shown below. It is a recent"
    \footnote \super \char##x00b6 \concat {
      \super \char##x00b6 \italic " Aug 2012"
    }
    "composition."
  }
  \relative c' {
    a'4 b8 e c4 d
  }
}

```

}

A simple tune [§] is shown below. It is a recent [¶] composition.



[§] *By me*
[¶] *Aug 2012*

Voir aussi

Manuel d'initiation : Section “Objets et interfaces” dans *Manuel d'initiation*.

Manuel de notation : [Commentaires textuels], page 213, [Équivalents ASCII], page 482, [Indications textuelles], page 216, [Info-bulle], page 207, Section A.12 [Liste des caractères spéciaux], page 686, [Unicode], page 481.

Référence des propriétés internes : Section “FootnoteEvent” dans *Référence des propriétés internes*, Section “FootnoteItem” dans *Référence des propriétés internes*, Section “FootnoteSpanner” dans *Référence des propriétés internes*, Section “Footnote_engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les notes de bas de page ne peuvent que s'empiler l'une au-dessus de l'autre ; elles ne seront jamais présentées à la queue leu leu.

Silence multimesures et ligatures automatiques ne peuvent se voir affecter de note de bas de page.

Les notes de bas de page peuvent générer des chevauchements quand elles sont trop nombreuses sur une même page.

3.2.4 Référencement des numéros de page

LilyPond vous permet, à l'aide de la commande `\label`, d'insérer des points de référence dans un ouvrage, aussi bien en dehors qu'au fil de la musique. Ce point de référence pourra être ensuite repris à l'intérieur d'un *markup* ; vous pourrez même y ajouter le numéro de page grâce à la commande de *markup* `\page-ref`.

```
\header { tagline = ##f }
\book {
  \label #'firstScore
  \score {
    {
      c'1
      \pageBreak \mark A \label #'markA
      c'1
    }
  }
}
```

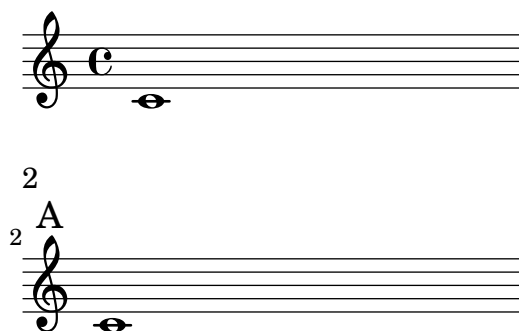


```

    }
}

\markup { Le premier mouvement débute à la page \page-ref #'firstScore "0" "?" }
\markup { Le repère A est à la page \page-ref #'markA "0" "?" }
}

```



Le premier mouvement débute à la page 1
Le repère A est à la page 2

L'instruction `\page-ref` prend trois arguments :

1. le point de référence, sous la forme d'un symbole Scheme, comme par exemple `#'firstScore`,
2. un « emporte-pièce » afin d'estimer la longueur totale du *markup*, et
3. un texte de remplacement au cas où la référence ne serait pas retrouvée.

La présence de l'emporte-pièce est rendue nécessaire par le fait que les *markups* sont générés avant que les sauts de page ne soient positionnés. Bien que le numéro de page en question ne soit pas encore déterminé, LilyPond doit connaître les dimensions de ce *markup*. Vous pouvez, lorsque l'ouvrage contiendra plus de dix pages, stipuler un emporte-pièce sur deux caractères – soit "00".

Commandes prédéfinies

`\label`, `\page-ref`.

3.2.5 Table des matières

La commande `\markuplist \table-of-contents` vous permettra de générer une table des matières. Les éléments qui la composeront sont créés par la commande `\tocItem`, insérée indépendamment ou au sein d'une expression musicale.

```

\markuplist \table-of-contents
\pageBreak

\tocItem \markup "Premier mouvement"
\score {
{
c'4 % ...
\tocItem \markup "Passage spécifique du premier mouvement"
d'4 % ...
}
}

```

```

    }
}

\tocItem \markup "Second mouvement"
\score {
  {
    e'4 % ...
  }
}

```

Les *markups* dévolus à la mise en forme de la table des matières se définissent dans le bloc `\paper`. Il s'agit par défaut de `tocTitleMarkup` pour le titre de la table, et de `tocItemMarkup` pour ses éléments – composés de leur libellé et numéro de page. Vous pouvez bien entendu personnaliser ces variables :

```

\paper {
  %% Translate the toc title into French:
  tocTitleMarkup = \markup \huge \column {
    \fill-line { \null "Table des matières" \null }
    \hspace #1
  }
  %% use larger font size
  tocItemMarkup = \markup \large \fill-line {
    \fromproperty #'toc:text \fromproperty #'toc:page
  }
}

```

Notez bien la manière de référencer le libellé et le numéro de page dans la définition de `tocItemMarkup`.

N'hésitez pas à définir vous-même d'autres commandes et *markups* afin de construire une table plus élaborée :

- commencez par définir une nouvelle variable de type `markup` au sein du bloc `\paper`,
- puis définissez une fonction musicale chargée d'insérer un élément de la table à partir de cette variable.

Dans l'exemple qui suit, nous avons créé un nouveau style d'élément dans le but de mentionner les actes dans la table des matières d'un opéra :

```

\paper {
  tocActMarkup = \markup \large \column {
    \hspace #1
    \fill-line { \null \italic \fromproperty #'toc:text \null }
    \hspace #1
  }
}

tocAct =
#(define-music-function (parser location text) (markup?)
  (add-toc-item! 'tocActMarkup text))

```

Table of Contents

Atto Primo

Coro. Viva il nostro Alcide	1
Cesare. Presti omai l'Egizzia terra	1

Atto Secondo

Sinfonia	1
Cleopatra. V'adoro, pupille, saette d'Amore	1

L'élément et son numéro de page peuvent se rejoindre par une ligne pointillée :

```
\header { tagline = ##f }
\paper {
  tocItemMarkup = \tocItemWithDotsMarkup
}

\book {
  \markuplist \table-of-contents
  \tocItem \markup { Allegro }
  \tocItem \markup { Largo }
  \markup \null
}
```

Table of Contents

Allegro	1
Largo	1

Commandes prédéfinies

`\table-of-contents`, `\tocItem`.

Voir aussi

Fichiers d'initialisation : `'../ly/toc-init.ly'`.

3.3 Travail sur des fichiers texte

3.3.1 Insertion de fichiers LilyPond

Lorsqu'un projet prend de l'importance en volume, il est judicieux de le scinder en plusieurs fichiers, auxquels vous ferez référence avec un simple

```
\include "autrefichier.ly"
```

Une ligne `\include "autrefichier.ly"` dans un fichier revient à recopier intégralement le contenu de `'autrefichier.ly'` à l'endroit même où est placée l'instruction `\include`. Vous pouvez par exemple écrire un fichier individuel par instrument, puis les regrouper pour former le fichier « conducteur ». Les différentes variables définies dans les fichiers séparés seront normalement reprises et utilisables dans le fichier formant le conducteur. Les sections balisées dans les fichiers individuels peuvent être réutilisées en différents endroits de la partition, comme expliqué à la rubrique [Section 3.3.2 \[Différentes éditions à partir d'une même source\]](#), page 475.

Lorsque le fichier auquel il est fait référence se trouve dans le même répertoire, donner seulement son nom en argument à la commande `\include` suffit. S'il se trouve ailleurs, vous devrez indiquer le chemin d'accès, absolu ou relatif, en respectant toutefois la syntaxe UNIX – autrement dit, le séparateur de répertoire est une oblique normale / et non l'oblique inverse \ de DOS ou Windows. Par exemple, si le fichier `'truc.ly'` se trouve dans le répertoire supérieur au répertoire de travail, la ligne devra être

```
\include "../truc.ly"
```

ou bien, si les fichiers correspondant aux parties d'orchestre se trouvent dans le sous-répertoire `'parties'` relativement au répertoire courant, vous devrez mentionner

```
\include "parties/VI.ly"
\include "parties/VII.ly"
... etc.
```

Les fichiers à inclure peuvent eux-mêmes contenir des instructions `\include`. Ces instructions `\include` de second niveau ne pourront, par défaut, être interprétées qu'une fois intégrées dans le fichier principal ; leur argument doit donc comporter le chemin relativement au fichier principal et non par rapport au fichier dans lequel cette inclusion est mentionnée. Vous pouvez toutefois influencer sur ce comportement de manière globale à l'aide de l'option `-drelative-includes` en ligne de commande ou en ajoutant une clause `#{ly:set-option 'relative-includes #t}` en tête du fichier principal.

Lorsque `relative-include` est valorisé à `#t`, le chemin à suivre pour chacune des commandes `\include` sera pris relativement au fichier qui la contient. Cette option est vouée à être activée par défaut dans une future version de LilyPond.

Selon l'endroit où `relative-includes` est valorisé à `#t` ou `#f`, la commande `\include` permettra d'incorporer des fichiers contenus dans l'arborescence du répertoire principal et des fichiers situés ailleurs. Si, par exemple, une bibliothèque générale `libA` a été créée pour utiliser des sous-fichiers à l'aide de d'inclusions dans un fichier catalogue, les clauses `\include` devront être précédées d'un `#{ly:set-option #relative-includes #t}` de telle sorte qu'elles soient interprétées correctement lorsque rapatriées dans le fichier `'libA.ly'` principal. Examinons cela dans les faits :

```
libA/
  libA.ly
  A1.ly
  A2.ly
  ...

puis le fichier catalogue, libA.ly, qui contient
#{ly:set-option 'relative-includes #t}
\include "A1.ly"
\include "A2.ly"
...
% retour au réglage par défaut
#{ly:set-option 'relative-includes #f}
```

Tout fichier `.ly` peut désormais consulter l'intégralité de cette bibliothèque grâce à un simple `\include "~/libA/libA.ly"`

Un positionnement judicieux des commutateurs permet de gérer des structures de fichiers plus complexes.

Vous pouvez inclure des fichiers dont vous spécifierez le chemin d'accès sur la ligne de commande au moment de lancer la compilation. L'appel à ces fichiers ne mentionnera alors que leur nom. Par exemple, si vous voulez compiler avec cette méthode le fichier `'principal.ly'`

qui inclut des fichiers situés dans le sous-répertoire ‘parties’, placez vous dans le répertoire contenant ‘principal.ly’, puis tapez

```
lilypond --include=parties principal.ly
    tout en ayant bien dans ‘principal.ly’
\include "VI.ly"
\include "VII.ly"
... etc
```

Lorsqu’un fichier est voué à être inclus dans nombre de partitions, vous pouvez le placer dans le répertoire de LilyPond ‘./ly’. Attention : ce répertoire varie selon votre installation, comme indiqué au chapitre [Section “Autres sources de documentation” dans Manuel d’initiation](#). Ce fichier sera inclus dès lors que vous fournirez uniquement son nom en argument à la fonction `\include`. C’est par exemple le cas du fichier de définition particulier ‘gregorian.ly’.

Au moment où vous lancez LilyPond, un certain nombre de fichiers se retrouvent inclus par défaut ; il suffit d’activer le mode verbeux en faisant `lilypond --verbose` pour s’en rendre compte. Vous verrez ainsi défiler, en plus de nombreuses informations, le nom d’un certain nombre de fichiers et de chemins d’accès. Les fichiers les plus importants sont mentionnés au chapitre [Section “Autres sources de documentation” dans Manuel d’initiation](#). Si vous venez à les modifier, rappelez-vous qu’ils seront écrasés à l’installation d’une nouvelle version de LilyPond.

Vous trouverez quelques exemples simples d’utilisation de la commande `\include` au chapitre [Section “Conducteurs et parties” dans Manuel d’initiation](#).

Voir aussi

Manuel d’initiation : [Section “Autres sources de documentation” dans Manuel d’initiation](#), [Section “Conducteurs et parties” dans Manuel d’initiation](#).

Problèmes connus et avertissements

Lorsque vous incluez un fichier qui porte le même nom que l’un des fichiers d’initialisation de LilyPond, le fichier de la distribution de LilyPond aura préséance.

3.3.2 Différentes éditions à partir d’une même source

Plusieurs méthodes permettent de générer différentes versions d’une partition à partir d’une même source. Les variables – ou identificateurs – sont sûrement le moyen le plus simple de combiner de différente manière des passages relativement longs, alors que les balises permettront de sélectionner de courts fragments selon leur utilisation.

Quelle que soit la méthode utilisée, séparer la notation de la structure de la partition vous donnera plus de liberté dans l’agencement de l’ouvrage final, puisque vous ne reviendrez pas sur la musique qui le compose.

Utilisation de variables

Un fragment musical identifié par une variable est réutilisable à divers endroits de la partition, comme nous l’avons vu à la rubrique [Section “Organisation du code source avec des variables” dans Manuel d’initiation](#). Par exemple, une partition pour chœur *a cappella* comporte souvent une réduction pour piano reprenant toutes les voix ; il s’agit de la même musique, et vous ne devrez donc la saisir qu’une seule fois. D’autre part, la musique issue de deux variables peut se combiner sur une seule portée, comme nous l’avons vu à la rubrique [\[Regroupement automatique de parties\], page 163](#). Prenons l’exemple suivant :

```
sopranoMusic = \relative c' { a4 b c b8( a) }
altoMusic = \relative g' { e4 e e f }
tenorMusic = \relative c' { c4 b e d8( c) }
```

```

bassMusic = \relative c' { a4 gis a d, }
allLyrics = \lyricmode {King of glo -- ry }
<<
  \new Staff = "Soprano" \sopranoMusic
  \new Lyrics \allLyrics
  \new Staff = "Alto" \altoMusic
  \new Lyrics \allLyrics
  \new Staff = "Tenor" {
    \clef "treble_8"
    \tenorMusic
  }
  \new Lyrics \allLyrics
  \new Staff = "Bass" {
    \clef "bass"
    \bassMusic
  }
  \new Lyrics \allLyrics
  \new PianoStaff <<
    \new Staff = "RH" {
      \set Staff.printPartCombineTexts = ##f
      \partcombine
      \sopranoMusic
      \altoMusic
    }
    \new Staff = "LH" {
      \set Staff.printPartCombineTexts = ##f
      \clef "bass"
      \partcombine
      \tenorMusic
      \bassMusic
    }
  >>
>>

```



Générer une partition chorale ou la réduction pour piano ne requiert que de modifier la structure des éléments, sans aucunement toucher à la musique.

Dans le cas d'une partition relativement longue, vous pouvez isoler la définition des différentes variables dans des fichiers séparés que vous appellerez ensuite, comme indiqué à la rubrique [Section 3.3.1 \[Insertion de fichiers LilyPond\]](#), page 473.

Utilisation de balises

La commande `\tag #'partieA` affecte à une expression musicale le nom *partieA*. Les expressions ainsi balisées pourront être filtrées par la suite, à l'aide de `\keepWithTag #'nom` ou `\removeWithTag #'nom`. Ces filtres fonctionnent de la manière suivante :

Filtre	Résultat
Musique balisée précédée de <code>\keepWithTag #'nom</code>	Musique non balisée et musique balisée par <i>nom</i> seront incluses ; la musique balisée autrement est exclue.
Musique balisée précédée de <code>\removeWithTag #'nom</code>	Musique non balisée et fragments appelés autrement que <i>nom</i> seront inclus ; la musique balisée par <i>nom</i> est exclue.
Musique balisée non précédée de <code>\keepWithTag</code> ou <code>\removeWithTag</code>	Musique balisée et non balisée seront incluses.

Les arguments des commandes `\tag`, `\keepWithTag` et `\removeWithTag` doivent être un symbole (tel que `#'conducteur` ou `#'partie`), suivi d'une expression musicale.

Dans l'exemple qui suit, nous obtenons deux versions du même extrait, l'une pour le conducteur, l'autre pour l'instrumentiste qui, elle, comportera les ornements développés.

```
music = \relative g' {
  g8. c32 d
  \tag #'trills { d8.\trill }
```

```

\tag #'expand { \repeat unfold 3 { e32 d } }
c32 d
}

\score {
  \keepWithTag #'trills \music
}
\score {
  \keepWithTag #'expand \music
}

```



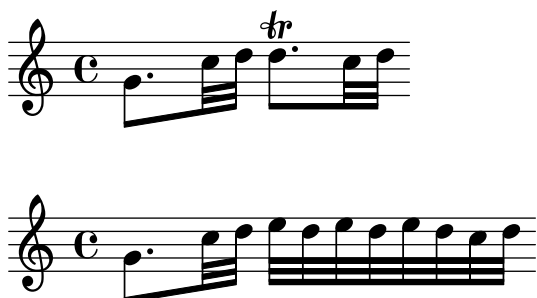
Il est parfois plus aisé d'exclure des fragments :

```

music = \relative g' {
  g8. c32 d
  \tag #'trills { d8.\trill }
  \tag #'expand {\repeat unfold 3 { e32 d } }
  c32 d
}

\score {
  \removeWithTag #'expand
  \music
}
\score {
  \removeWithTag #'trills
  \music
}

```



Ce principe de filtrage peut s'appliquer aux articulations, textes etc. Il suffit de positionner `-\tag #ma-balise` avant l'articulation ou le texte, comme ici :


```
c1-\tag #'doigt ^4
c1-\tag #'gaffe ^"Attention !"
```

Ceci définira une note avec une indication conditionnelle de doigté ou un texte.

Vous pouvez baliser différemment la même expression musicale en saisissant plusieurs `\tag` :

```
music = \relative c'' {
  \tag #'a \tag #'both { a4 a a a }
  \tag #'b \tag #'both { b4 b b b }
}
<<
\keepWithTag #'a \music
\keepWithTag #'b \music
\keepWithTag #'both \music
>>
```



L'application concomitante de plusieurs filtres `\removeWithTag` à la même expression musicale permet d'exclure plusieurs balisages :

```
music = \relative c'' {
  \tag #'A { a4 a a a }
  \tag #'B { b4 b b b }
  \tag #'C { c4 c c c }
  \tag #'D { d4 d d d }
}
{
  \removeWithTag #'B
  \removeWithTag #'C
  \music
}
```



L'application de plus d'un filtre `\keepWithTag` à la même expression musicale aboutit à l'exclusion de **tous** les balisages. En effet, si le premier filtre exclut tous les autres balisages, l'application du second exclura les effets du premier.

Il peut arriver que vous ayez besoin de raccorder quelque chose en un point particulier d'une expression musicale. Les commandes `\pushToTag` et `\appendToTag` permettent d'insérer du matériel, qu'il soit antérieur ou postérieur, à des **éléments** d'une construction musicale existante. La musique séquentielle ou simultanée comporte assurément des **éléments** :

L'apparence réelle des caractères est déterminée par les glyphes ou graphèmes tels que définis dans les différentes polices disponibles. Une police, ou une fonte, définit la mise en correspondance d'un sous-ensemble de points de code unicode en glyphes. LilyPond recourt à la bibliothèque Pango pour assurer le rendu des textes multilingues.

LilyPond n'effectue aucune conversion d'encodage que ce soit. Ceci implique donc que tout texte – un titre, des paroles ou même une instruction musicale – comportant des caractères non ASCII soit codé en UTF-8. Le plus sûr moyen de saisir du texte de la sorte consiste à utiliser un éditeur supportant l'unicode et à enregistrer vos fichiers en UTF-8. C'est le cas pour la plupart des éditeurs actuels, que ce soit vim, Emacs, jEdit et GEdit. Tous les systèmes Windows postérieurs à NT utilisent Unicode en natif ; même Notepad est capable d'éditer et sauvegarder un fichier en UTF-8 – sans parler de l'excellente alternative qu'est BabelPad.

La compilation d'un fichier LilyPond comportant des caractères non ASCII qui n'aurait pas été enregistré dans l'encodage UTF-8 vous renverra l'erreur

```
FT_Get_Glyph_Name () erreur : invalid argument
```

Voici un exemple utilisant du texte en cyrillique, en hébreux et en portugais.



Unicode

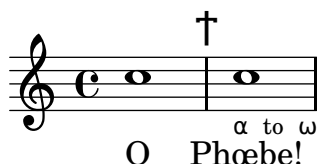
Lorsque vous avez besoin d'un caractère dont vous connaissez le point de code mais que votre éditeur ne permet pas de saisir directement, vous pouvez utiliser les instructions `\char ##xhhhh` ou `\char #dddd` au sein d'un bloc `\markup – hhhh` et `dddd` correspondant respectivement à la valeur hexadécimale ou décimale. Même s'il est inutile de saisir les zéros superflus, il est de bon ton de stipuler les quatre caractères formant la représentation hexadécimale. Évitez cependant l'encodage UTF-8 d'un point de code après un `\char` ; les encodages UTF-8 comprennent un bit supplémentaire indiquant le nombre d'octets. Une table de correspondance entre les codes Unicode et le nom des caractères ainsi que leur code hexadécimal est disponible sur le site du consortium Unicode, <http://www.unicode.org/>.

Par exemple, `\char ##x03BE` et `\char #958` correspondent tous deux au caractère unicode U+03BE, dénommé « Greek Small Letter Xi ».

Quel que soit le point de code spécifié de cette manière, il ne vous sera alors pas nécessaire d'enregistrer votre fichier en UTF-8. Vous devrez toutefois disposer d'une fonte contenant ce caractère qui soit accessible à LilyPond.

L'exemple suivant illustre la manière d'insérer un caractère sous sa forme hexadécimale, à la fois dans un repère, dans une articulation, dans des paroles et dans du texte indépendant.

```
\score {
  \relative c' {
    c1 \mark \markup { \char ##x03EE }
    c1_\markup { \tiny { \char ##x03B1 " to " \char ##x03C9 } }
  }
  \addlyrics { 0 \markup { \concat { Ph \char ##x0153 be! } } }
}
\markup { "Copyright 2008--2012" \char ##x00A9 }
```



Copyright 2008--2012 ©

Le signe *copyright* dans le champ de titrage consacré s'inscrit de la manière suivante :

```
\header {
  copyright = \markup { \char ##x00A9 "2008" }
}
```

Équivalents ASCII

Dès lors que vous aurez inclus la liste de leur équivalent ASCII, LilyPond reconnaîtra un certain nombre de caractères spéciaux :

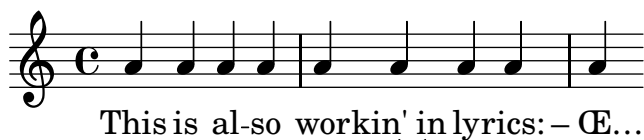
```
\paper {
  #(include-special-characters)
}

\markup "&flqq; &ndash; &OE;uvre incomplète&hellip; &frqq;"

\score {
  \new Staff { \repeat unfold 9 a'4 }
  \addlyrics {
    This is al -- so wor -- kin'~in ly -- rics: &ndash;_&OE;&hellip;
  }
}

\markup \column {
  "The replacement can be disabled:"
  "&ndash; &OE; &hellip;"
  \override #'(replacement-alist . ()) "&ndash; &OE; &hellip;"
}
```

« – Œuvre incomplète... »



The replacement can be disabled:

– Œ ...

– &OE; …

L'extension de cette liste est possible aussi bien de manière globale :

```
\paper {
  #(add-text-replacements!
    '(("100" . "hundred")
      ("dpi" . "dots per inch")))
}

\markup "A 100 dpi."
```

A hundred dots per inch.

qu'en un point particulier de votre source :

```
\markup \replace #'(("100" . "hundred")
                      ("dpi" . "dots per inch")) "A 100 dpi."
```

A hundred dots per inch.

Voir aussi

Manuel de notation : [Section A.12 \[Liste des caractères spéciaux\]](#), page 686.

Fichiers d'initialisation : 'ly/text-replacements.ly'.

3.4 Contrôle des sorties

3.4.1 Extraction de fragments musicaux

LilyPond permet d'extraire des fragments d'une partition à l'instar du choriste amateur qui alimente son album avec des coupures de journaux.

Vous devrez pour cela définir les mesures à découper. La définition suivante, incluse dans votre fichier source,

```
\layout {
  clip-regions
  = #(list
    (cons
      (make-rhythmic-location 5 1 2)
      (make-rhythmic-location 7 3 4)))
}
```

vous permettra d'extraire un fragment compris entre le milieu de la cinquième mesure et quelque part dans la septième. Le triplet 5 1 2 signifie : après la durée d'une blanche dans la mesure 5 ; le 7 3 4 signifie quant à lui que l'on s'arrête à la mesure 7, après la durée de trois noires.

Rien ne vous empêche d'extraire plusieurs fragments, dès lors que vous définissez dans la liste d'autres paires d'emplacements rythmiques.

Cette fonctionnalité n'est toutefois effective que lorsque vous lancez LilyPond avec l'option `-dclip-systems`. Les « coupures » seront générées sous la forme de fichiers EPS, convertis en PDF ou PNG selon le format que vous aurez stipulé.

Pour de plus amples informations quant au format des résultats, consultez le chapitre [Section “Lancement de lilypond”](#) dans *Utilisation des programmes*.

3.4.2 Ignorer des passages de la partition

Dans un travail de transcription ou de recopie de la musique, ce qui vous intéresse plus particulièrement se situe à la fin, là même où vous en êtes dans la notation. Dans le but de gagner du temps dans le processus de correction, vous pouvez « escamoter » le reste et ne générer que les dernières mesures en insérant

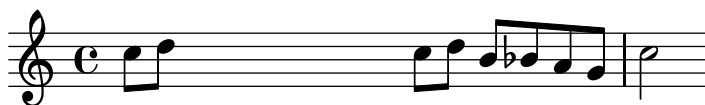
```
showLastLength = R1*5
\score { ... }
```

dans votre fichier source. Ceci aura pour effet de ne générer que les cinq dernières mesures – si tant est que le morceau soit à 4/4 – de tous les `\score` de votre fichier. Dans le cas d'un œuvre conséquente, cette pratique s'avère fort utile puisqu'elle évite de tout générer. Vous pourriez aussi être amené à retravailler le début d'une œuvre, pour y ajouter une partie par exemple, auquel cas c'est la propriété `showFirstLength` que vous utiliserez.

Vous pouvez contrôler très finement les parties à escamoter, grâce au commutateur `Score.skipTypesetting` : lorsqu’il est activé, aucune gravure n’est réalisée.

Ce commutateur agit aussi sur la sortie MIDI. Notez bien que tous les événements seront escamotés, y compris les changements de tempo ou d’instrument – vous voilà prévenu !

```
c8 d
\set Score.skipTypesetting = ##t
e8 e e e e e e e
\set Score.skipTypesetting = ##f
c8 d b bes a g c2
```



Dans le cadre de musique polyphonique, `Score.skipTypesetting` s’applique à toutes les voix et portées. Vous gagnerez donc encore plus de temps.

3.4.3 Formats de sortie alternatifs

En matière de partition imprimable, LilyPond génère par défaut des documents au format PostScript (PS) et Portable Document Format (PDF). Vous pouvez aussi obtenir des documents au format Scalable Vector Graphics (SVG), Encapsulated PostScript (EPS) ou Portable Network Graphics (PNG) dès lors que vous aurez lancé LilyPond en ligne de commande avec l’option *ad hoc* – voir [Section “Utilisation en ligne de commande”](#) dans *Utilisation des programmes* à ce sujet.

3.4.4 Changement des fontes musicales

Gonville est une alternative à la fonte Feta que LilyPond utilise par défaut. Vous pouvez la télécharger à partir de

<http://www.chiark.greenend.org.uk/~sgtatham/gonville/>

Voici quelques mesures utilisant la police Gonville :



Et ces même mesures avec la police de LilyPond, Feta :



Instructions d’installation pour MacOS

Téléchargez puis décompressez l’archive zip. Recopiez le répertoire `lilyfonts` dans `‘SHARE_DIR/lilypond/current’` – voir [Section “Autres sources de documentation”](#) dans *Manuel d’initiation* à ce sujet. Renommez le répertoire `fonts` qui s’y trouve en `fonts_orig`, puis le répertoire `lilyfonts` en `fonts`. Il vous suffira, pour retrouver la fonte Feta, de renommer `fonts_orig` en `fonts`.

Voir aussi

Manuel d’initiation : [Section “Autres sources de documentation”](#) dans *Manuel d’initiation*.

Problèmes connus et avertissements

Gonville ne permet pas de générer de la notation ancienne, et certains glyphs ajoutés depuis lors aux jeux de caractères en soient absent. Consultez le site de l'auteur pour de plus amples informations ainsi qu'à propos des conditions d'utilisation.

3.5 Sortie MIDI

MIDI (Musical Instrument Digital Interface) constitue un standard en matière de connexion et de contrôle des instruments électroniques. Un fichier MIDI contient une série de notes réparties dans différentes pistes. Il ne s'agit en rien d'un fichier sonore ; il vous faudra un logiciel capable de traduire ces séries de notes en sons réels.

Vous pouvez convertir vos partition en fichier MIDI de manière à entendre ce que vous avez saisi. Ceci vous permet de contrôler aisément ce que vous avez saisi : octaves et altérations erronées heurteront votre oreille avertie grâce au MIDI.

Le fichier MIDI généré par LilyPond est relativement brut. Vous pouvez cependant obtenir un meilleur rendu avec [Section 3.5.7 \[Le script Articulate\], page 494](#).

Dans une sortie MIDI, LilyPond alloue un canal à chaque portée, tout en réservant le canal 10 aux percussions. Dans la mesure où un périphérique MIDI ne comprend que 16 canaux, un fichier MIDI qui comportera plus de quinze portées verra le même canal réutilisé.

3.5.1 Création de fichiers MIDI

LilyPond générera un fichier MIDI dès que vous ajouterez un bloc `\midi` à la structure de votre partition, comme ici :

```
\score {
  ...musique...
  \midi { }
}
```

Lorsque le bloc `\score` contient un bloc `\midi` mais pas de bloc `\layout`, LilyPond ne produira qu'une sortie MIDI. Si donc vous avez besoin aussi d'un support visuel, vous devrez aussi mentionner un bloc `\layout`.

```
\score {
  ...musique...
  \midi { }
  \layout { }
}
```

Hauteurs, durées, liaisons de prolongation, nuances et changements de tempo seront interprétés et traduits en événements MIDI. Les indications de nuances, crescendos et decrescendos sont traduits en niveau de volume ; les indications sous la forme d'une fraction déterminée du volume disponible, et crescendos et decrescendos sous la forme d'une progression linéaire entre les deux extrêmes. Le rendu des indications de nuance peut être désactivé pour le MIDI – voir [Section 3.5.2 \[Le bloc MIDI\], page 487](#).

Le tempo initial ainsi que ses changements sont normalement indiqués au fil de la notation à l'aide de la commande `\tempo` ; ils seront retranscrits dans le fichier MIDI. La commande `\tempo` donne lieu à l'impression d'une indication métronomique que vous pouvez toutefois rendre invisible, comme indiqué à la rubrique [\[Indication métronomique\], page 63](#). Une autre manière de spécifier le tempo initial pour un fichier MIDI est indiquée plus avant – voir [Section 3.5.2 \[Le bloc MIDI\], page 487](#).

En raison de certaines limitations de Windows, les fichiers MIDI doivent y porter l'extension `.mid`. D'autres systèmes utilisent l'extension `.midi`. Si besoin est, placez la ligne suivante au début de votre fichier source, avant l'ouverture de tout bloc `\book`, `\bookpart` ou `\score` :

```
#(ly:set-option 'midi-extension "midi")
```

Cette ligne déterminera .midi comme extension par défaut pour les fichiers MIDI.

Vous pouvez aussi le faire en ligne de commande :

```
lilypond ... -dmidi-extension=midi fichierLily.ly
```

Noms d'instrument

L'instrument MIDI affecté à un canal particulier est déterminé par la propriété `Staff.midiInstrument`. Vous pouvez choisir l'un des instruments répertoriés à l'annexe [Section A.6 \[Instruments MIDI\]](#), page 618.

```
\new Staff {
  \set Staff.midiInstrument = #"glockenspiel"
  ...notes...
}
\new Staff \with {midiInstrument = #"cello"} {
  ...notes...
}
```

Lorsque l'instrument choisi ne correspond pas exactement à l'une des dénominations consacrées, LilyPond le remplacera par un piano de concert ("`acoustic grand`").

Morceaux choisis

Affectation d'un canal MIDI par voix

Lorsque LilyPond génère un fichier MIDI, chaque portée sera par défaut affectée à un canal, quel que soit le nombre de voix qu'elle contient. Ceci permet d'éviter de se retrouver à court de canaux, sachant qu'il n'y en a que seize de disponibles.

Le fait de déplacer le `Staff_performer` dans le contexte `Voice` permet d'affecter à chaque voix d'une même portée un canal MIDI spécifique. Dans l'exemple suivant, la même portée donnera lieu à deux canaux MIDI différents, chacun étant affecté de son propre `midiInstrument`.

```
\score {
  \new Staff <<
    \new Voice \relative c'' {
      \set midiInstrument = #"flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = #"clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi { }
```



```

\context {
  \Staff
  \remove "Staff_performer"
}
\context {
  \Voice
  \consists "Staff_performer"
}
\tempo 2 = 72
}
}

```



Problèmes connus et avertissements

Un changement de volume ne peut intervenir que sur le démarrage d'une note. C'est la raison pour laquelle la succession d'un crescendo et d'un diminuendo ne peut se produire sur une même note.

Certains lecteurs MIDI ne rendent pas correctement les changements de tempo. MS Windows Media Player et **timidity** le font sans problème.

3.5.2 Le bloc MIDI

Dès lors que vous désirez obtenir une sortie MIDI, vous devrez inscrire un bloc `\midi` au sein du bloc `\score`. Son fonctionnement est comparable à ce lui du bloc `\layout`, voire plus simple. Si le bloc `\midi` est la plupart du temps laissé vide, il peut contenir des aménagements pour certains contextes, la définition de contextes particuliers ou du code permettant de déterminer la valeur de certaines propriétés. L'exemple suivant détermine le tempo initial du fichier MIDI tout en se passant de son indication sur la partition imprimée.

```

\score {
  ...musique...
  \midi {
    \tempo 4 = 72
  }
}

```

Ici, le tempo est fixé à 72 noires à la minute. Spécifier un tempo de la sorte ne permet pas de donner une valeur pour une note pointée. Vous devrez, en pareil cas, subdiviser la note pointée en durée plus courte. Par exemple, indiquer 90 à la noire pointée est équivalent à spécifier 270 croches à la minute :

```
tempoWholesPerMinute = #(ly:make-moment 270 8)
```

La syntaxe permettant de définir un contexte pour le `\midi` est en tout point identique à celle que vous utilisez dans le bloc `\layout`, à ceci près que le « graveur » est remplacé par un « interprète ». Les différents contextes disponibles en matière de MIDI sont répertoriés dans le fichier d'initialisation `../ly/performer-init.ly` – pour plus de détail, voir [Section “Autres sources de documentation”](#) dans *Manuel d'initiation*. Si vous voulez vous passer des nuances dans votre fichier MIDI, il vous suffit d'insérer les lignes suivantes dans votre bloc `\midi{ }`.

```
\midi {
  ...
  \context {
    \Voice
    \remove "Dynamic_performer"
  }
}
```

LilyPond ne générera de sortie MIDI que si vous incluez un bloc `\midi` dans la structure de la partition, initialisée par la commande `\score`.

```
\score {
  { ...notes... }
  \midi { }
}
```

3.5.3 Contenu de la sortie MIDI

Éléments pris en compte dans le MIDI

Un fichier MIDI généré par LilyPond comprendra les éléments de notation suivants :

- les hauteurs ;
- les microtonalités – voir [\[Altérations\]](#), page 5. Leur rendu nécessite cependant un lecteur qui prenne en charge la modulation ;
- les accords nommés ;
- le rythme en tant que durée de note, y compris les nolets ;
- les trémolos, exceptés ceux utilisant la syntaxe « `:[nombre]` » ;
- les liaisons de prolongation ;
- les indications de nuance ;
- les crescendos et decrescendos s'étalant sur plusieurs notes ;
- les changements de tempo indiqués par un `\tempo` ;
- les paroles.

Si vous utilisez [Section 3.5.7 \[Le script Articulate\]](#), page 494, d'autres éléments seront alors inclus :

- articulations (lié, staccato, etc.),
- trilles et groupettos,
- rallentando et accelerando.

Éléments non pris en compte dans le MIDI

LilyPond ne peut générer d'événement MIDI pour les éléments suivant, sauf à utiliser [Section 3.5.7 \[Le script Articulate\]](#), page 494 :

- le rythme indiqué sous forme d'annotation (par ex. *swing*) ;
- les changements de tempo indiqués sous forme d'annotation (sans `\tempo`) ;
- les staccatos et autres articulations ou ornements ;
- les liaisons d'articulation et de phrasé ;
- les crescendos ou decrescendos sur une seule note ;
- les trémolos indiqués par la syntaxe « `:[nombre]` » ;
- la basse chiffrée ;
- les accords en microtonalité.

3.5.4 Répétitions et MIDI

Au prix de quelques réglages, les reprises de toutes sortes peuvent être rendues dans le fichier MIDI. Il suffit pour cela de recourir à la fonction `\unfoldRepeats`, qui développe toutes les reprises. En d'autres termes, `\unfoldRepeats` transforme toutes les reprises en reprises de type `unfold`.

```
\unfoldRepeats {
  \repeat tremolo 8 {c'32 e' }
  \repeat percent 2 { c''8 d'' }
  \repeat volta 2 {c'4 d' e' f'}
  \alternative {
    { g' a' a' g' }
    {f' e' d' c' }
  }
}
\bar "|."
```



Dans une partition comportant plusieurs voix, le développement des reprises ne sera effectif en MIDI qu'à la condition que ces reprises soient mentionnée correctement dans **toutes** les voix.

Lorsque l'on veut utiliser `\unfoldRepeats` seulement pour le rendu MIDI, il faut établir deux blocs `\score` : un pour le MIDI, avec des reprises explicites, et l'autre pour la partition, avec des reprises notées sous forme de barres de reprise, de trémolo ou de symboles de pourcentage. Par exemple

```
\score {
  ..musique..
  \layout { .. }
}
\score {
  \unfoldRepeats ..musique..
  \midi { .. }
}
```

3.5.5 Gestion des nuances en MIDI

Les nuances MIDI sont générées par le `Dynamic_performer`, affecté par défaut au contexte `Voice`. Vous pouvez contrôler à la fois le volume général, celui des indications de nuance ainsi que celui des différents instruments.

Indications de nuance

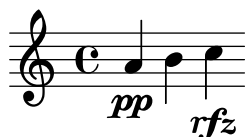
Les indications de nuances sont transcrites en fraction du volume MIDI. Nous allons, par défaut, de 0,25 pour un *ppppp* à 0,95 pour un *ffff*. Les correspondances entre nuance et fraction de volume sont répertoriées dans le fichier '`../scm/midi.scm`' – consultez la rubrique [Section "Autres sources de documentation" dans Manuel d'initiation](#) si vous ne savez comment le localiser. Vous pouvez modifier ou étendre ce jeu grâce à une fonction qui prendra en argu-

ment une indication de nuance et renverra la fraction désirée, puis en affectant cette fonction à `Score.dynamicAbsoluteVolumeFunction`.

Prenons un exemple. Votre partition comporte un *rinforzando* que vous avez indiqué par `\rfz`. Cette indication de nuance n'étant pas répertoriée dans le jeu par défaut, elle ne produira aucun effet en MIDI. Il en sera d'ailleurs de même pour toute indication créée de toute pièce à l'aide de l'instruction `make-dynamic-script`. Voici comment procéder pour ajuster le volume MIDI de ce *rinforzando* que le compositeur a indiqué. La fonction `Scheme` définit une fraction de 0,9 en cas de *rfz*, et demande d'utiliser la fonction par défaut dans les autres cas.

```
#(define (myDynamics dynamic)
  (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = #"cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative c'' {
        a4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
}
```



Si vous étiez amené à devoir modifier l'intégralité du jeu des correspondances, nous vous conseillons d'utiliser la procédure `default-dynamic-absolute-volume` contenue dans le fichier `'../scm/midi.scm'` ainsi que la table d'association comme base. Le dernier exemple de cette partie illustre la manière de procéder.

Amplitude du volume en MIDI

Les valeurs extrêmes du volume MIDI des nuances se contrôlent à l'aide des propriétés `midiMinimumVolume` et `midiMaximumVolume` qui agissent au niveau `Score`. Ces propriétés sont effectives dès lors qu'une nuance est indiquée ; une nuance de départ est donc requise pour que le volume soit ajusté dès le début de la partition. Vous pouvez alors modifier la fraction correspondant à chaque nuance à l'aide de la formule

`midiMinimumVolume + (midiMaximumVolume - midiMinimumVolume) * fraction`

Voici comment ajuster les nuances tout en limitant l'amplitude du volume entre 0,2 et 0,5 :

```
\score {
  <<
  \new Staff {
    \key g \major
    \time 2/2
    \set Staff.midiInstrument = #"flute"
```

```

\new Voice \relative c''' {
  r2 g\mp g fis~
  fis4 g8 fis e2~
  e4 d8 cis d2
}
}
\new Staff {
  \key g \major
  \set Staff.midiInstrument = #"clarinet"
  \new Voice \relative c'' {
    b1\p a2. b8 a
    g2. fis8 e
    fis2 r
  }
}
>>
\layout {}
\midi {
  \context {
    \Score
    tempoWholesPerMinute = #(ly:make-moment 72 2)
    midiMinimumVolume = #0.2
    midiMaximumVolume = #0.5
  }
}
}

```



Égalisation de plusieurs instruments (i)

La définition de l'amplitude du volume MIDI au niveau d'un contexte **Staff** permet de gérer les volumes relatifs entre les différents instruments. Ceci constitue en quelque sorte un égaliseur, ce qui permet d'améliorer notablement la qualité de la sortie MIDI.

La clarinette de l'exemple suivant jouera relativement moins fort que la flûte. Rappelez-vous que pour que cela fonctionne correctement, la première note de chacun des instruments doit être affublée d'une indication de nuance.

```

\score {
  <<
  \new Staff {
    \key g \major
    \time 2/2
    \set Staff.midiInstrument = #"flute"
    \set Staff.midiMinimumVolume = #0.7
    \set Staff.midiMaximumVolume = #0.9
    \new Voice \relative c''' {

```

```

        r2 g\mp g fis~
        fis4 g8 fis e2~
        e4 d8 cis d2
    }
}
\new Staff {
    \key g \major
    \set Staff.midiInstrument = #"clarinet"
    \set Staff.midiMinimumVolume = #0.3
    \set Staff.midiMaximumVolume = #0.6
    \new Voice \relative c'' {
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
    }
}
>>
\layout {}
\midi {
    \tempo 2 = 72
}
}

```



Égalisation de plusieurs instruments (ii)

Lorsque les propriétés volume minimum et maximum n'ont pas été définies, LilyPond appliquera par défaut une légère égalisation pour quelques instruments. Les instruments concernés ainsi que le niveau d'égalisation sont répertoriés dans une table *instrument-equalizer-alist* du fichier `'../scm/midi.scm'`.

Vous pouvez remplacer l'égaliseur basique en définissant une nouvelle procédure Scheme `instrumentEqualizer` au sein du contexte `Score`. Cette procédure prend en unique argument le nom d'un instrument MIDI et renverra une paire de fractions correspondant au minimum et maximum de volume alloué à cet instrument. Cette substitution fonctionne selon le même principe que celui que nous avons vu en début de chapitre avec `dynamicAbsoluteVolumeFunction`. L'égaliseur par défaut, *default-instrument-equalizer*, est défini dans le fichier `'../scm/midi.scm'` ; son analyse vous aidera à construire votre propre procédure.

Nous allons, dans l'exemple suivant, régler le volume relatif de la flûte et de la clarinette – au même niveau que dans le précédent.

```

#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(

```

```

    ("flute" . (0.7 . 0.9))
    ("clarinet" . (0.3 . 0.6)))
  my-instrument-equalizer-alist))

#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = #"flute"
      \new Voice \relative c''' {
        r2 g\mp g fis~
        fis4 g8 fis e2~
        e4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \new Voice \relative c'' {
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {
    \tempo 2 = 72
  }
}

```



3.5.6 MIDI et percussions

La notation pour percussions recourt généralement à un contexte particulier – le **DrumStaff** – qui permet d'affecter directement les instruments concernés au canal 10 qui leur est réservé.

Certains instruments, tels le xylophone, le marimba, le vibraphone ou les timbales, se traitent cependant comme des instruments « classiques » puisqu'ils sont capables d'émettre des hauteurs

différentes ; leurs notation relève donc d'un contexte **Staff** standard, et non d'un **DrumStaff** pour pouvoir être rendus correctement en MIDI.

D'autres percussions, bien que n'émettant qu'un seul son et inclus dans le standard *general MIDI*, comme le tom mélodique ou le tambour taiko, ne sont pas attachés au canal 10. Ces instruments doivent donc être saisi dans un contexte **Staff**, en utilisant la hauteur appropriée.

De nombreux instruments de la famille des percussions, les castagnettes par exemple, n'existent pas dans le standard *general MIDI*. L'alternative, bien que peu satisfaisante, consiste à leur attribuer le son le plus proche dans la banque standard.

Problèmes connus et avertissements

Le standard *general MIDI* ne dispose pas du *rim shot* ; LilyPond lui substitue un *sidestick*.

3.5.7 Le script Articulate

Vous obtiendrez un rendu MIDI plus « réaliste » grâce au script **articulate**. Celui-ci va tout faire pour d'une part prendre en compte les articulations (liaisons, staccato etc.) – en ajoutant un blanc aux notes raccourcies – et, d'autre part, développer les trilles ou groupettos ainsi que tenir compte des éventuels rallentando et accelerando.

L'utilisation du script **articulate** se fait en deux temps. Vous devez dans un premier temps inclure son fichier d'initialisation en ajoutant en tête de votre fichier la ligne

```
\include "articulate.ly"
```

puis, dans le bloc **\score**, indiquer que toutes les répétitions seront développées et appliquer la commande à votre musique :

```
\unfoldRepeats \articulate <<
tout le reste du bloc contenant la partition...
>>
```

Une fois votre fichier modifié de la sorte, vous constaterez que la version imprimable aura été modifiée en profondeur. Le bloc **\midi** produira par contre un fichier MIDI de bien meilleure qualité.

Bien que cela ne gêne en rien le fonctionnement du script **articulate**, lui adjoindre la commande **\unfoldRepeats** comme illustré ci-dessus permettra le rendu d'un certain nombre d'articulations tels les trilles.

Problèmes connus et avertissements

Dans la mesure où le script **articulate** tend à raccourcir les accords, certaines musiques, notamment pour l'orgue, paraîtront de moins bonne qualité.

3.6 Extraction d'informations musicales

En plus de générer du graphisme et du MIDI, LilyPond peut présenter l'information musicale sous forme textuelle.

3.6.1 Affichage de notation au format LilyPond

La fonction musicale **\displayLilyMusic** permet d'afficher en notation LilyPond une expression musicale. Le résultat défilera dans le terminal après avoir lancé LilyPond en ligne de commande. Par exemple,

```
{
  \displayLilyMusic \transpose c a, { c4 e g a bes }
}
```

affichera


```
{ a,4 cis e fis g }
```

LilyPond affichera le résultat sous forme de message en console, au milieu de toutes les informations de compilation. Afin d'isoler ces messages et enregistrer le résultat de la fonction `\display{MATÉRIAU}`, pensez à rediriger la sortie vers un fichier.

```
lilypond fichier.ly > affichage.txt
```

Vous noterez que LilyPond ne se contente pas de simplement afficher l'expression musicale, mais procède aussi à son interprétation – du fait que `\displayLilyMusic` renvoie l'expression tout en l'affichant. S'il est bien pratique d'insérer un `\displayLilyMusic` dans une expression musicale pour en obtenir des informations, l'interprétation de cette expression peut toutefois être évitée en ajoutant un `\void` avant l'instruction :

```
{
  \void \displayLilyMusic \transpose c a, { c4 e g a bes }
}
```

3.6.2 Affichage de la musique sous forme d'expression Scheme

Voir [Section “Affichage d'expressions musicales”](#) dans *Extension de LilyPond*.

3.6.3 Enregistrement d'événements musicaux dans un fichier

LilyPond vous permet de sauvegarder dans un fichier séparé, sur la base de la portée, les événements musicaux. Vous devrez pour ce faire inclure dans votre fichier maître un fichier d'initialisation spécifique :

```
\include "event-listener.ly"
```

Pour chaque portée que comporte votre partition, vous obtiendrez un fichier ‘`NOMFICHIER-PORTÉE.NOMMÉE.notes`’ ou ‘`NOMFICHIER-unnamed-staff.notes`’. Notez bien que si plusieurs portées ne sont pas explicitement nommées, tous leurs événements seront regroupés et mélangés dans le même fichier. Le résultat ressemblera à ceci :

```
0.000  note      57      4  p-c 2 12
0.000  dynamic  f
0.250  note      62      4  p-c 7 12
0.500  note      66      8  p-c 9 12
0.625  note      69      8  p-c 14 12
0.750  rest      4
0.750  breathe
```

Il s'agit d'un tableau dont les colonnes sont délimitées par une tabulation. Chaque ligne comporte deux champs fixes suivis d'un certain nombre de paramètres optionnels.

```
temps type ...paramètres...
```

Ces informations peuvent faire l'objet d'un retraitement par d'autres programmes, comme des scripts python, aux fins de recherche en analyse musicologique ou des expériences à partir du rendu de LilyPond.

Problèmes connus et avertissements

Tous les événements ne sont pas pris en charge par ‘`event-listener.ly`’. Il s'agit en premier lieu d'une démonstration, un « proof of concept » du potentiel de LilyPond. Si certains des éléments que vous cherchez à obtenir n'apparaissent pas, recopiez le fichier ‘`event-listener.ly`’ dans votre répertoire et modifiez-le de telle sorte qu'il travaille selon vos attentes.

4 Gestion de l'espace

L'agencement général d'une partition dépend de trois facteurs interdépendants : la mise en page, les sauts de ligne et l'espacement. Les choix faits en matière d'espacement détermineront la densité de chacun des systèmes, ce qui influera sur le positionnement des sauts de ligne et, par voie de conséquence, sur le nombre de pages de la partition.

En pratique, cette procédure comporte quatre étapes. Dans un premier temps, des distances élastiques (*springs*) sont déterminées sur la base des durées. Sont alors calculées différentes possibilités de saut de ligne, chacune se voyant attribuer un « coefficient de laideur », puis est estimée la hauteur de chaque système. LilyPond opte enfin pour la combinaison entre sauts de page et de ligne qui offre la meilleure occupation de l'espace, tant horizontalement que verticalement.

Les réglages qui influencent la mise en forme se placent dans deux blocs différents. Le bloc `\paper {...}` étudié à la rubrique [Section 4.1 \[Mise en forme de la page\], page 496](#) contient les réglages applicables à l'intégralité du document – à toutes les partitions d'un ouvrage – tels que format du papier, impression ou non des numéros de page, etc. Quant au bloc `\layout {...}`, qui fait l'objet de la rubrique [Section 4.2 \[Mise en forme de la partition\], page 507](#), il détermine la mise en forme de la musique : le nombre de systèmes utilisés, l'espacement des regroupements de portées, etc.

Note : Vous verrez au fil de ce chapitre apparaître certains termes dont la traduction vous semblera assurément erronée. Il n'en est cependant rien : certains termes techniques ont une histoire particulière selon leur langue d'origine. Ainsi le vocable anglais *Ragged* signifie en lambeau, en loques ; dans l'univers typographique, un maître français voit un alignement à gauche – il dira « au fer à gauche » – alors que son homologue anglophone constate un *ragged-right* – donc du vide à droite.

4.1 Mise en forme de la page

Nous allons examiner ici les options qui contrôlent la mise en forme des pages attachées au bloc `\paper`.

4.1.1 Le bloc `\paper`

Un bloc `\paper` peut apparaître au sein d'un bloc `\book`. Les réglages effectués à l'aide du bloc `\paper` s'appliqueront à l'intégralité d'un ouvrage, qui peut comporter plusieurs partitions (des blocs *score*) ; il ne doit donc en aucun cas se trouver dans un bloc `\score`. Peuvent apparaître dans un bloc `\paper` :

- la fonction Scheme `set-paper-size`,
- des variables propres au bloc `\paper` qui viendront adapter la mise en page,
- la définition des différents *markups* qui personnaliseront la mise en forme des entêtes et pieds de page ainsi que des titrages.

La fonction `set-paper-size` fait l'objet de la rubrique qui suit – [Section 4.1.2 \[Format du papier et adaptation automatique\], page 497](#). Les variables du bloc `\paper` chargées de la mise en page sont abordées plus loin dans ce chapitre. Quant aux définitions relatives aux *markups* des entête, pied de page et titrage, elles sont étudiées à la rubrique [Section 3.2.2 \[Titrages personnalisés\], page 458](#).

La plupart des variables gérant le papier ne sont fonctionnelles que lorsque mentionnées dans un bloc `\paper`. Certaines, qui peuvent toutefois apparaître dans un bloc `\layout`, sont référencées à la rubrique [Section 4.2.1 \[Le bloc `\layout`\], page 507](#).

Sauf mention contraire, toutes les variables du bloc `\paper` qui correspondent à des dimensions sont exprimées en millimètre – vous pouvez bien entendu spécifier un autre système de mesure. Voici comment, par exemple, définir la marge haute (`top-margin`) à dix millimètres :

```
\paper {
  top-margin = 10
}
```

Si vous préférez lui affecter une valeur de 0,5 pouce, vous devrez mentionner le suffixe d'unité `\in` :

```
\paper {
  top-margin = 0.5\in
}
```

LilyPond accepte les suffixes d'unité `\mm`, `\cm`, `\in` et `\pt`. Ces unités sont des conversions de millimètres, répertoriées dans le fichier `'ly/paper-defaults-init.ly'`. Pour plus de lisibilité, et bien que ce ne soit pas techniquement requis, nous vous conseillons d'ajouter `\mm` à votre code lorsque vous travaillez en millimètres.

Vous pouvez aussi définir les valeurs du bloc `\paper` à l'aide de fonctions Scheme. Voici l'équivalent de l'exemple précédent :

```
\paper {
  #(define top-margin (* 0.5 in))
}
```

Voir aussi

Manuel de notation : [Section 4.1.2 \[Format du papier et adaptation automatique\]](#), page 497, [Section 3.2.2 \[Titrages personnalisés\]](#), page 458. [Section 4.2.1 \[Le bloc \layout\]](#), page 507.

Fichiers d'initialisation : `'ly/paper-defaults-init.ly'`.

4.1.2 Format du papier et adaptation automatique

Format du papier

LilyPond génère par défaut, et en l'absence de mention explicite d'un format de papier, un fichier imprimable au format A4. Vous pouvez cependant utiliser un autre format à l'aide des deux fonctions `set-default-paper-size` et `set-paper-size` qui, elle, se place à l'intérieur d'un bloc `\paper` :

```
 #(set-default-paper-size "quarto")
    qui se place en début de fichier, ou set-paper-size,
\paper {
  #(set-paper-size "tabloid")
}
```

qui s'inscrit au sein d'un bloc `\paper`.

La seule restriction à l'utilisation isolée de la fonction `set-default-paper-size` est qu'elle doit intervenir avant le premier bloc `\paper`. `set-default-paper-size` fixe le format pour toutes les pages, alors que `set-paper-size` détermine le format des feuilles rattachées à un bloc `\paper` particulier. Ainsi, lorsque le bloc `\paper` se trouve en tête de fichier, le format du papier s'appliquera à toutes les pages ; si `\paper` apparaît dans un bloc `\book`, la taille ne s'appliquera qu'au *book* en question.

À l'intérieur d'un bloc `\paper`, la fonction `set-paper-size` doit intervenir avant toute autre variable. Les raisons à ceci sont abordées à la rubrique [\[Adaptation automatique au format\]](#), page 498.

Différents formats de papier sont définis dans le fichier `'scm/paper.scm'`. Bien que vous puissiez y ajouter votre propre format, sachez cependant que celui-ci est écrasé à chaque mise à jour de LilyPond.

La commande suivante, inscrite dans votre fichier, vous permettra d'ajouter votre format personnalisé à ceux déjà connus, puis d'y faire appel à l'aide des fonctions `set-default-paper-size` et `set-paper-size` :

```
#(set! paper-alist (cons '("mon format" . (cons (* 15 in) (* 3 in))) paper-alist))■
```

```
\paper {
  #(set-paper-size "mon format")
}
```

Les unités peuvent s'exprimer aussi bien en `in` (pouces), qu'en `cm` (centimètres) ou `mm` (millimètres).

Le fait d'ajouter l'argument `'landscape` à l'instruction stipulant le format du papier permet d'obtenir une présentation à l'italienne – ou paysage si vous préférez – et donc des lignes plus longues.

```
#(set-default-paper-size "a6" 'landscape)
```

L'inversion des dimensions du papier sans pour autant basculer la présentation – comme pour imprimer sur une carte postale ou créer un graphique destiné à inclusion – s'obtient en ajoutant `'landscape` au nom du format de papier :

```
#(set-default-paper-size "a6landscape")
```

Lorsque la taille du papier comporte explicitement `'landscape` ou `'portrait`, la présence d'un argument `'landscape` aura pour seul effet de modifier l'orientation de l'image et non les dimensions de la feuille.

Voir aussi

Manuel de notation : [\[Adaptation automatique au format\]](#), page 498, [Section A.5 \[Formats de papier prédéfinis\]](#), page 616.

Fichiers d'initialisation : `'scm/paper.scm'`.

Adaptation automatique au format

Toute modification du format de papier à l'aide des fonctions Scheme `set-default-paper-size` ou `set-paper-size`, que nous avons vues à la rubrique [\[Format du papier\]](#), page 497, se traduira automatiquement par l'ajustement d'un certain nombre de variables attachées au bloc `\paper` afin qu'elles soient en concordance avec le format spécifié. Vous pouvez annuler l'ajustement automatique d'une variable particulière en redéfinissant sa valeur après avoir spécifié le format de papier utilisé. Notez bien que le simple fait d'affecter une valeur à `paper-height` ou `paper-width` ne déclenchera pas l'étalement automatique, bien que spécifier une largeur de papier (`paper-width`) peut influencer d'autres valeurs – mais c'est une autre histoire dont nous parlerons plus tard et qui n'a rien à voir avec la mise à l'échelle.

L'adaptation automatique affecte les dimensionnements verticaux `top-margin` et `bottom-margin` – voir [Section 4.1.3 \[Variables d'espacement vertical fixe\]](#), page 499 –, ainsi que les dimensionnements horizontaux `left-margin`, `right-margin`, `inner-margin`, `outer-margin`, `binding-offset`, `indent` et `short-indent` – voir [Section 4.1.5 \[Variables d'espacement horizontal\]](#), page 502.

Les valeurs par défaut de ces dimensionnements sont contenues dans le fichier `'ly/paper-defaults-init.ly'` et utilisent les variables internes `top-margin-default`, `bottom-margin-default`, etc. correspondant au format par défaut – papier A4 – pour lequel `paper-height` est à 297\mm et `paper-width` à 210\mm.

Voir aussi

Manuel de notation : [Section 4.1.3 \[Variables d'espacement vertical fixe\]](#), page 499, [Section 4.1.5 \[Variables d'espacement horizontal\]](#), page 502.

Fichiers d'initialisation : `'ly/paper-defaults-init.ly'`, `'scm/paper.scm'`.

4.1.3 Variables d'espacement vertical fixe

Note : Certains dimensionnements attachés au bloc `\paper` sont automatiquement ajustés selon le format du papier, ce qui peut conduire à un résultat inattendu – voir [\[Adaptation automatique au format\]](#), page 498.

Les valeurs par défaut (avant étalonnage) sont définies dans le fichier `'ly/paper-defaults-init.ly'`.

`paper-height`

La hauteur de la feuille. Il s'agit par défaut de la dimension du papier utilisé. Notez bien que cette variable n'affectera pas l'ajustement automatique d'un certain nombre de dimensionnements verticaux.

`top-margin`

La marge entre le bord supérieur de la feuille et la surface imprimable. Elle est fixée par défaut à `5\mm` et s'ajustera selon le format de papier.

`bottom-margin`

La marge entre la surface imprimable et le bord inférieur de la feuille. Elle est fixée par défaut à `6\mm` et s'ajustera selon le format de papier.

`ragged-bottom`

L'activation de cette variable permet de ne pas répartir verticalement les systèmes sur les pages hormis la dernière. La valeur par défaut est `#f`. Lorsque la partition ne comporte que deux ou trois systèmes par page, comme pour un conducteur d'orchestre, nous vous conseillons d'activer cette variable.

`ragged-last-bottom`

La désactivation de cette variable permet de répartir verticalement les systèmes de la dernière page d'une partition. La valeur par défaut est `#t`. Nous vous conseillons, lorsque des pièces couvrent deux pages ou plus, de désactiver cette variable. Notez bien que la variable `ragged-last-bottom` affecte aussi la dernière page de chacune des parties – créées à l'aide d'un bloc `\bookpart` – d'un même ouvrage.

Voir aussi

Manuel de notation : [\[Adaptation automatique au format\]](#), page 498.

Fichiers d'initialisation : `'ly/paper-defaults-init.ly'`.

Morceaux choisis : [Section “E spacements” dans Morceaux choisis](#).

Problèmes connus et avertissements

Les titrages (contenus dans le bloc `\header{}`) sont considérés comme des systèmes à part entière ; ils seront donc affectés par `ragged-bottom` et `ragged-last-bottom`, qui éventuellement ajouteront de l'espace avant le premier système de la partition.

La définition explicite d'un format de papier annulera tout réglage des marges haute et basse.

4.1.4 Variables d'espacement vertical fluctuant

Il est souvent judicieux d'apporter un peu de flexibilité à l'espacement entre différents éléments (marges, titres, systèmes ou mouvements), en dilatation ou compression selon le cas. Un certain nombre de variables de type `\paper` répertoriées ci-dessous vous permettront d'affiner ces dimensionnements.

Gardez à l'esprit que les variables du bloc `\paper` dont nous parlons ici n'influencent en rien l'espacement des portées d'un même système. L'espacement au sein des systèmes est géré par des propriétés attachées à des objets graphiques (*grobs*) qui, elles, se définissent au niveau du bloc `\score` – voir à ce sujet [Section 4.4.1 \[Espacement vertical au sein d'un système\]](#), page 518.

Structure des variables d'espacement vertical fluctuant

Chacune de ces variables attachées au bloc `\paper` est constituée d'une liste associative (*alist*) à quatre *clés* :

- **basic-distance** (*distance de base*) – la grandeur d'espace par défaut, exprimée en hauteur de portée, séparant les *points de référence* de deux éléments, qui évite tout risque de collision en l'absence de dilatation ou compression. Le point de référence d'un titre ou d'un *markup* est son sommet, celui d'un système est le centre vertical du `StaffSymbol` le plus proche – même lorsqu'une ligne de « non-portée » viendrait à s'intercaler. Une **basic-distance** inférieure à **padding** ou **minimum-distance** sera sans effet, dans la mesure où l'espacement résultant ne saurait être inférieur à **padding** ou **minimum-distance**.
- **minimum-distance** (*distance-minimale*) – l'espacement minimal, exprimé en hauteur de portée, entre les points de référence des deux éléments alors qu'il y a déjà un effet de compression. Une **minimum-distance** inférieure à la valeur du **padding** sera sans effet, dans la mesure où l'espacement résultant ne saurait être inférieur au **padding**.
- **padding** (*décalage*) – la grandeur minimale de « blanc » qui sépare deux éléments, exprimée en hauteur de portée. On peut le voir comme la hauteur minimale d'un rectangle vide qui devrait s'étendre sur toute la largeur des deux éléments.
- **stretchability** (*dilatation*) – le coefficient d'étirement de cet espace. Un coefficient nul permet de figer l'espacement, à moins qu'il n'en résulte des collisions. Un coefficient positif déterminera la propension d'un espacement à s'étirer, tout en tenant compte du coefficient affecté aux autres espacements. Par exemple, lorsque le coefficient de dilatation d'une dimension est double de celui d'une autre, elle pourra s'étirer deux fois plus que cette dernière. Il ne saurait être négatif. La valeur `+inf.0` provoque une `programming_error` (erreur de programmation) et est ignorée ; vous pouvez toutefois utiliser `1.0e7` pour obtenir une valeur proche de l'infini. Lorsque cette *clé* n'est pas définie, sa valeur est par défaut égale à **space**. Notez bien que l'utilisateur ne peut définir une propension à la compression ; elle est en fait égale à (**basic-distance** – **minimum-distance**).

Lorsque l'impression n'est pas en pleine page – elle est donc *ragged bottom* pour les anglophones – l'élément **space** n'est pas étiré. Les hauteurs sur une telle page correspondront donc au maximum de

- **basic-distance**, plus
- **minimum-distance** et
- **padding**, augmenté de ce qu'il faut pour éviter les chevauchements.

Cependant, lorsque la partition fait plusieurs pages, la dernière page reprendra dans la mesure du possible l'espacement de la page précédente.

Les manières de modifier des listes associatives font l'objet d'un [Section “chapitre spécifique” dans Manuel de notation](#). L'exemple suivant indique deux façons de modifier une liste associative. La première déclaration intervient sur une seule clé, alors que la deuxième redéfinit complètement la variable.

```
\paper {
  system-system-spacing #'basic-distance = #8

  score-system-spacing =
    #'((padding . 1)
      (basic-distance . 12)
      (minimum-distance . 6)
      (stretchability . 12))
}
```

Liste des variables d'espacement fluctuant

Le nom des dimensionnements à hauteur variable sont de la forme *avant-après-spacing*, où *avant* et *après* représentent les éléments qui doivent être espacés. La distance s'établit entre les points de référence des deux éléments concernés (voir la rubrique précédente pour plus de précision). Notez bien que, dans les règles de nommage des variables qui suivent, le terme *markup* fait référence aussi bien à un *markup de titrage* (`bookTitleMarkup` ou `scoreTitleMarkup`) qu'à un *markup de haut niveau* (voir [Section 3.1.5 \[Structure de fichier\], page 449](#)). Toutes les distances sont exprimées en espace de portée.

Leurs valeurs par défaut sont inscrites dans le fichier `'ly/paper-defaults-init.ly'`.

markup-system-spacing
détermine l'espacement entre un titre ou un *markup* de premier niveau, et le système qui le suit.

score-markup-spacing
détermine l'espacement entre le dernier système et le titre ou *markup* de haut niveau qui le suit.

score-system-spacing
détermine l'espacement entre le dernier système d'une partition et le premier système de la partition suivante, en l'absence de titrage ou *markup* qui les sépare.

system-system-spacing
détermine l'espacement entre deux systèmes d'un même mouvement.

markup-markup-spacing
détermine l'espacement entre deux titres ou *markups* de premier niveau.

last-bottom-spacing
détermine la distance entre le dernier système ou le dernier *markup* de haut niveau, et le bas de la surface imprimable – autrement dit le haut de la marge basse.

top-system-spacing
détermine l'espace entre le haut de la surface imprimable (le bas de la marge haute) et le milieu du premier système. Cette variable n'est effective qu'en l'absence de titre ou *markup* de premier niveau en haut de page.

top-markup-spacing
détermine l'espace entre le haut de la surface imprimable (le bas de la marge haute) et le premier titre ou *markup* de premier niveau. Cette variable n'est effective qu'en l'absence de système en haut de page.

Voir aussi

Manuel de notation : [Section 4.4.1 \[Espacement vertical au sein d'un système\], page 518](#).
 Morceaux choisis : [Section “Espacements” dans *Morceaux choisis*](#).
 Morceaux choisis : [Section “Espacements” dans *Morceaux choisis*](#).

4.1.5 Variables d'espacement horizontal

Note : Certains dimensionnements attachés au bloc `\paper` sont automatiquement ajustés selon le format du papier, ce qui peut conduire à un résultat inattendu – voir [\[Adaptation automatique au format\]](#), page 498.

Variables de marge et de largeur

Les valeurs par défaut (avant étalonnage) sont définies dans le fichier `'ly/paper-defaults-init.ly'`.

`paper-width`

La largeur de la page. Elle correspond par défaut à la largeur du format de papier utilisé. Si `paper-width` n'a aucun effet en matière d'ajustement automatique, cette variable influe sur la variable `line-width`. Lorsque vous définissez à la fois les valeurs de `paper-width` et `line-width`, les valeurs de `left-margin` et `right-margin` seront recalculées. Voir aussi `check-consistency`.

`line-width`

la longueur d'un système musical sans indentation et justifié sur toute la largeur de la page. La valeur par défaut est égale à `paper-width`, auquel sont retranchés `left-margin` et `right-margin`. Lorsque vous définissez `line-width` sans modifier les valeurs de `left-margin` et `right-margin`, les marges seront alors recalculées de telle sorte que les systèmes soient centrés. Voir aussi `check-consistency`. La variable `line-width` peut se définir aussi dans un bloc `\layout`.

`left-margin`

la marge entre le bord gauche de la feuille et le début de chaque système. La valeur par défaut est de `10\mm` ; elle sera ajustée selon le format du papier. Lorsque vous définissez `line-width` et `right-margin` sans modifier la valeur de `left-margin`, cette dernière sera alors égale à $(\text{paper-width} - \text{line-width} - \text{right-margin})$. Lorsque seule `line-width` est définie, les deux marges correspondent à $((\text{paper-width} - \text{line-width}) / 2)$, ce qui a pour effet de centrer les systèmes sur la page. Voir aussi `check-consistency`.

`right-margin`

La marge entre le bord droit de la page et la fin des systèmes en pleine largeur (non *ragged*). La valeur par défaut est de `10\mm` et s'ajustera selon le format du papier. Lorsque vous définissez `line-width` et `left-margin`, sans modifier la valeur de `right-margin`, cette dernière sera alors égale à $(\text{paper-width} - \text{line-width} - \text{left-margin})$. Lorsque seule `line-width` est définie, les deux marges correspondent à $((\text{paper-width} - \text{line-width}) / 2)$, ce qui a pour effet de centrer les systèmes sur la page. Voir aussi `check-consistency`.

`check-consistency`

Lorsqu'elle est activée, cette variable vérifie que `left-margin`, `line-width` et `right-margin` sont en cohérence, et que l'addition de ces trois éléments ne dépassera pas la largeur du papier (`paper-width`). La valeur par défaut est `\#t`. Dans le cas d'une incohérence, un message d'avertissement est émis et les trois variables – marges et longueur de ligne – rétablies à leur valeur par défaut (ajustées selon le format du papier).

`ragged-right`

Lorsque cette variable est activée, les systèmes ne s'étendront pas sur la longueur de la ligne, mais s'arrêteront à leur longueur normale. La valeur par défaut est `\#f`.

mais, si la partition ne comporte qu'un seul système, elle passe à **#t**. Cette variable peut aussi se gérer au sein d'un bloc `\layout`.

ragged-last

Lorsqu'elle est activée, cette variable permet de ne pas étendre le dernier système de façon à occuper toute la longueur de la ligne. La valeur par défaut est **#f**. Cette variable peut aussi se gérer au sein d'un bloc `\layout`.

Voir aussi

Manuel de notation : [\[Adaptation automatique au format\]](#), page 498.

Fichiers d'initialisation : `'ly/paper-defaults-init.ly'`.

Problèmes connus et avertissements

La définition explicite d'un format de papier annulera tout réglage des marges gauche et droite.

Variables spécifiques pour l'impression recto-verso

Les valeurs par défaut (avant étalonnage) sont définies dans le fichier `'ly/paper-defaults-init.ly'`.

two-sided

Cette variable permet de gérer efficacement les impressions recto-verso. Lorsqu'elle est activée, les réglages affectés à **inner-margin**, **outer-margin** ainsi que **binding-offset** détermineront les différentes marges selon qu'il s'agit d'une page paire ou impaire. Cette variable s'applique en lieu et place de **left-margin** et **right-margin**. La valeur par défaut est **#f**.

inner-margin

La marge que toutes les pages d'une partie ou de tout un ouvrage devront avoir du côté intérieur. Bien entendu, cette variable n'est effective que lorsque vous comptez générer un fichier imprimable en recto-verso – propriété **two-sided** définie à vrai. La valeur par défaut est de `10\mm` et s'ajustera selon le format du papier.

outer-margin

la marge que toutes les pages d'une partie ou de tout un ouvrage devront avoir du côté extérieur – opposé à la reliure. Bien entendu, cette variable n'est effective que lorsque vous comptez générer un fichier imprimable en recto-verso – propriété **two-sided** définie à vrai. La valeur par défaut est de `20\mm` et s'ajustera selon le format du papier.

binding-offset

La gouttière, ou marge de reliure, permet d'augmenter en conséquence la valeur de la marge intérieure **inner-margin** de telle sorte que rien ne soit masqué par la reliure. Bien entendu, cette variable n'est effective que lorsque vous comptez générer un fichier imprimable en recto-verso – propriété **two-sided** définie à vrai. La valeur par défaut est de 0 et s'ajustera selon le format du papier.

Voir aussi

Manuel de notation : [\[Adaptation automatique au format\]](#), page 498.

Fichiers d'initialisation : `'ly/paper-defaults-init.ly'`.

Variables d'indentation et de décalage

Les valeurs par défaut (avant étalonnage) sont définies dans le fichier `'ly/paper-defaults-init.ly'`.

horizontal-shift

Tous les systèmes, ainsi que les titres et séparateurs de systèmes, seront poussés d'autant vers la droite. La valeur par défaut est de 0.0.

indent

Le niveau d'indentation du premier système d'une partition. La valeur par défaut est de 15\mm et s'ajustera selon le format du papier. Cette variable peut aussi se gérer au sein d'un bloc `\layout`.

short-indent

Le niveau d'indentation de tous les systèmes hormis le premier. La valeur par défaut est de 0, et s'ajustera selon le format du papier dès lors que vous lui affecterez une valeur. Cette variable peut aussi se gérer au sein d'un bloc `\layout`.

Voir aussi

Manuel de notation : [\[Adaptation automatique au format\]](#), page 498.

Fichiers d'initialisation : `'ly/paper-defaults-init.ly'`.

Morceaux choisis : [Section "E spacements"](#) dans *Morceaux choisis*.

4.1.6 Autres variables du bloc `\paper`**Variables de gestion des sauts de ligne****max-systems-per-page**

Le nombre maximal de systèmes qu'une page pourra comporter. Cette variable n'est prise en compte, à ce jour, que par l'option `ly:optimal-breaking`, et n'est pas définie.

min-systems-per-page

Le nombre minimal de systèmes qu'une page pourra comporter. Attention cependant aux risques de débordement s'il est trop important. Cette variable n'est prise en compte, à ce jour, que par l'option `ly:optimal-breaking`, et n'est pas définie.

systems-per-page

Le nombre de systèmes que devrait comporter chaque page. Cette variable n'est à ce jour prise en charge que par l'algorithme `ly:optimal-breaking` et n'est pas définie par défaut.

system-count

Le nombre de systèmes requis par la partition. Cette variable n'est pas définie par défaut. Cette variable peut se gérer au sein d'un bloc `\layout`.

Voir aussi

Manuel de notation : [Section 4.3.1 \[Sauts de ligne\]](#), page 510.

Variables de gestion des sauts de page

Les valeurs par défaut sont définies dans le fichier `'ly/paper-defaults-init.ly'`.

blank-after-score-page-force

Pénalité pour apparition d'une page blanche entre deux partitions. Sa valeur est par défaut inférieure à celle de `blank-page-force` ; nous préférons qu'une page blanche s'insère après la fin de la partition plutôt qu'au milieu.

blank-last-page-force

Pénalité pour fin de partition intervenant sur une page impaire. La valeur par défaut est de 0.

blank-page-force

Pénalité pour apparition d'une page blanche en cours de partition. L'algorithme `ly:optimal-breaking` n'en tiendra pas compte puisqu'il ne conçoit pas la présence d'une page blanche au milieu d'une partition. La valeur par défaut est de 5.

page-breaking

L'algorithme de calcul des sauts de page à utiliser. Vous avez le choix entre `ly:minimal-breaking`, `ly:page-turn-breaking` et `ly:optimal-breaking`.

page-breaking-system-system-spacing

Cette variable permet de « tromper » l'algorithme de gestion des sauts de page quant à la valeur de `system-system-spacing`. Ainsi, lorsque `page-breaking-system-system-spacing #'padding` a une valeur nettement supérieure à `system-system-spacing #'padding`, l'algorithme en question aura tendance à disposer moins de systèmes sur une même page. Cette variable est par défaut non définie.

page-count

Le nombre de pages que devra comporter la partition. Cette variable est par défaut non définie.

Voir aussi

Manuel de notation : [Section 4.3.2 \[Sauts de page\]](#), page 512, [Section 4.3.3 \[Optimisation des sauts de page\]](#), page 513, [Section 4.3.4 \[Optimisation des tournes\]](#), page 513, [Section 4.3.5 \[Minimisation des sauts de page\]](#), page 514, [Section 4.3.6 \[Présentation en rouleau\]](#), page 514.

Fichiers d'initialisation : `'ly/paper-defaults-init.ly'`.

Variables de gestion des numéros de page

Les valeurs par défaut sont définies dans le fichier `'ly/paper-defaults-init.ly'`.

auto-first-page-number

L'algorithme qui gère les sauts de page prend en compte le fait que le premier numéro de page soit pair ou impair. Lorsque cette fonctionnalité est activée, l'algorithme des sauts de page décidera de lui-même si le premier numéro sera pair ou impair, ce qui se traduira par un éventuel incrément de un. La valeur par défaut est `#f`.

first-page-number

Le numéro de la première page. La valeur par défaut est de `#1`.

print-first-page-number

Cette variable permet d'imprimer le numéro de page y compris sur la première. La valeur par défaut est `#f`.

print-page-number

La désactivation de cette variable permet d'obtenir des pages non numérotées. La valeur par défaut est `#t`.

Voir aussi

Fichiers d'initialisation : `'ly/paper-defaults-init.ly'`.

Problèmes connus et avertissements

Les pages au numéro impair sont toujours à droite. Pour que la musique commence en page 1, le dos de la page de garde doit être vide de telle sorte que la page une se retrouve à droite.

Variables supplémentaires

page-spacing-weight

Cette variable définit l'importance relative des espacements entre la page (verticalité) et la ligne (horizontalité). Une valeur élevée privilégiera l'espacement au niveau de la page. La valeur par défaut est de 10.

print-all-headers

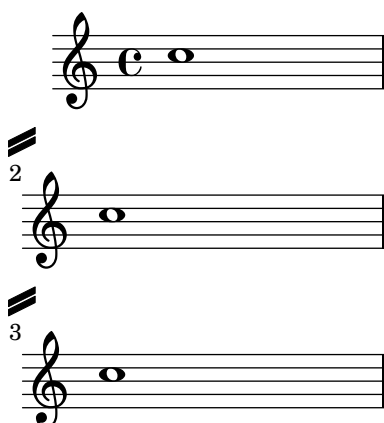
Lorsque cette variable est activée, l'intégralité des champs d'entête sera imprimée pour chaque bloc `\score`, plutôt que les seuls champs `piece` et `opus`. La valeur par défaut est `#f`.

system-separator-markup

Il s'agit en l'occurrence d'insérer un objet de type *markup* entre chaque système, comme on le voit dans nombre de partitions orchestrales. Cette variable n'est pas définie par défaut. La commande `\slashSeparator` – définie dans le fichier `'ly/titling-init.ly'` – fournit un *markup* relativement courant :

```
#(set-default-paper-size "a8")

\book {
  \paper {
    system-separator-markup = \slashSeparator
  }
  \header {
    tagline = ##f
  }
  \score {
    \relative c'' { c1 \break c1 \break c1 }
  }
}
```



Voir aussi

Fichiers d'initialisation : `'ly/titling-init.ly'`.

Morceaux choisis : [Section “Espacements” dans *Morceaux choisis*](#).

Problèmes connus et avertissements

L'entête par défaut, formé d'une seule ligne, est constitué du numéro de page et du champ `instrument` contenu dans le bloc `\header`.

4.2 Mise en forme de la partition

Nous allons voir ici les options du bloc `\layout`. Elles sont plus particulièrement destinées à gérer la mise en forme de la partition.

4.2.1 Le bloc `\layout`

Alors que le bloc `\paper` définit le formatage des pages pour l'intégralité du document, le bloc `\layout` s'occupe de la mise en forme spécifique à la partition. La mise en forme de la musique peut concerner toutes les partitions d'un même ouvrage, auquel cas un bloc `\layout` indépendant se placera en tête de fichier. Dans le cas où la mise en forme concerne une partition en particulier, un bloc `\layout` se placera au sein du bloc `\score` en question. Sont susceptibles d'apparaître dans un bloc `\layout` :

- la fonction Scheme `layout-set-staff-size`,
- dans des blocs `\context`, les modifications apportées aux différents contextes, et
- les variables normalement attachées au bloc `\paper` qui affecteront la mise en forme de la partition.

La fonction `layout-set-staff-size` fait l'objet de la rubrique suivante, [Section 4.2.2 \[Définition de la taille de portée\], page 509](#). La modification des contextes est abordée dans d'autres chapitres – voir [Section 5.1.4 \[Modification des greffons de contexte\], page 555](#) et [Section 5.1.5 \[Modification des réglages par défaut d'un contexte\], page 557](#). Les variables du bloc `\paper` que l'on peut retrouver dans un bloc `\layout` sont :

- `line-width`, `ragged-right` et `ragged-last` (voir [\[Variables de marge et de largeur\], page 502](#))
- `indent` et `short-indent` (voir [\[Variables d'indentation et de décalage\], page 503](#))
- `system-count` (voir [\[Variables de gestion des sauts de ligne\], page 504](#))

Voici un exemple de bloc `\layout` :

```
\layout {
  indent = 2\cm
  \context {
    \StaffGroup
    \override StaffGrouper #'staff-staff-spacing #'basic-distance = #8
  }
  \context {
    \Voice
    \override TextScript #'padding = #1
    \override Glissando #'thickness = #3
  }
}
```

Il est tout à fait possible que plusieurs blocs `\layout` cohabitent en tant qu'expressions de niveau supérieur. Ceci se révèle particulièrement utile lorsque différents réglages sont stockés dans des fichiers séparés qui sont inclus au besoin. Lorsqu'un bloc `\layout` est évalué, une copie de la configuration du `\layout` actuel est réalisée en interne, augmentée des aménagements apportés. Bien qu'on puisse considérer que le contenu des différents blocs `\layout` se cumule, c'est la dernière adaptation qui aura préséance en cas de situation conflictuelle – cas typique d'une même propriété modifiée dans différents blocs.

Par exemple, placer le bloc suivant

```
\layout {
  \context {
    \Voice
```

```

\override TextScript #'color = #magenta
\override Glissando #'thickness = #1.5
}
}

```

après celui de l'exemple précédent aura pour effet de cumuler les adaptations de `'padding` et `'color` pour l'objet `TextScript`, mais la dernière adaptation apportée à la propriété `'thickness` de `Glissando` remplace, ou masque, celle précédemment établie.

Les blocs `\layout` peuvent faire l'objet de variables, aux fins de les utiliser ultérieurement. Ceci requiert toutefois une attention particulière dans la mesure où cette manière de procéder n'est pas équivalente à une définition complète et globale.

Lorsque nous définissons la variable suivante,

```

layoutVariable = \layout {
  \context {
    \Voice
    \override NoteHead #'font-size = #4
  }
}

```

qui contient une configuration de `\layout` avec l'adaptation `NoteHead #'font-size`, cette combinaison n'est pas enregistrée en tant que configuration courante. Notez bien que la « configuration courante » est lue lorsque la variable est définie, non lorsqu'elle est utilisée ; par voie de conséquence, le contenu de la variable dépend de l'endroit où elle se trouve dans le code source.

Notre variable peut alors être utilisée au sein d'un autre bloc `\layout`, comme par exemple :

```

\layout {
  \layoutVariable
  \context {
    \Voice
    \override NoteHead #'color = #red
  }
}

```

Un bloc `\layout` qui contient une variable comme ci-dessus, ne recopie pas la configuration actuelle ; il utilise en fait le contenu de `layoutVariable` en tant que configuration de base pour les adaptations ultérieures, en conséquence de quoi toute modification intervenant entre la définition et l'utilisation de la variable sera perdue.

Si `layoutVariable` est définie, ou rapatriée par un `\include`, juste avant d'être utilisée, son contenu devient la configuration actuelle augmentée des adaptations que la variable contient. Considérant l'exemple d'utilisation de `layoutVariable` ci-dessus, le bloc `\layout` final contiendra donc :

```

TextScript #'padding = #1
TextScript #'color = #magenta
Glissando #'thickness = #1.5
NoteHead #'font-size = #4
NoteHead #'color = #red

```

ainsi que les adaptations de `indent` et `StaffGroup`.

Cependant, si la variable avait été définie bien avant le premier bloc `\layout`, la configuration actuelle ne contiendrait que

```

NoteHead #' font-size= #4 % (écrit dans la définition de la variable)
NoteHead #' color = #red % (ajouté après l'utilisation de la variable)

```

Une gestion attentive des variables de `\layout` se révèle être un outil précieux dans la mise en forme des sources et le retour à une configuration donnée.

Voir aussi

Manuel de notation : [Section 5.1.5 \[Modification des réglages par défaut d'un contexte\]](#), page 557.

Morceaux choisis : [Section “E spacements” dans *Morceaux choisis*](#).

4.2.2 Définition de la taille de portée

La **taille de portée** (*staff size*) est fixée par défaut à 20 points. Il existe deux manières de la modifier :

La taille des portées peut se définir globalement pour toutes les partitions d'un même fichier, ou plus précisément d'un bloc `\book`, à l'aide de `set-global-staff-size`.

```
 #(set-global-staff-size 14)
```

Ceci définit donc la hauteur des portées à 14 points par défaut ; toutes les fontes seront ajustées en conséquence.

Vous pouvez aussi spécifier une taille à une partition en particulier en procédant comme ci-dessous :

```
 \score{
   ...
   \layout{
     #(layout-set-staff-size 15)
   }
 }
```

La fonte Feta fournit les symboles musicaux dans huit tailles différentes. Chaque fonte correspond à une hauteur particulière de portée ; les petites tailles comportent des symboles plus épais pour être cohérent avec l'épaisseur relativement plus importante des lignes de la portée. Le tableau suivant répertorie les différentes tailles de police.

nom de la fonte	hauteur de portée (pt)	hauteur de portée (mm)	utilisation
feta11	11,22	3,9	format de poche
feta13	12,60	4,4	
feta14	14,14	5,0	
feta16	15,87	5,6	
feta18	17,82	6,3	carnet de chant
feta20	20	7,0	partition standard
feta23	22,45	7,9	
feta26	25,2	8,9	

Ces fontes sont disponibles dans toutes les tailles. La propriété de contexte `fontSize` ainsi que la propriété de mise en forme `staff-space` (voir [Section “StaffSymbol” dans *Référence des propriétés internes*](#)) permettent d'ajuster individuellement la taille de chaque portée. La taille de chacune des portées est relative à la taille globale.

Voir aussi

Manuel de notation : [\[Indication de la taille de fonte musicale\]](#), page 200.

Morceaux choisis : [Section “E spacements” dans *Morceaux choisis*](#).

Problèmes connus et avertissements

`layout-set-staff-size` ne modifie en rien l'espacement entre les lignes d'une portée.

4.3 Sauts

4.3.1 Sauts de ligne

Les sauts de ligne sont normalement gérés de façon automatique. Ils interviennent de telle sorte qu'une ligne ne soit ni trop resserrée, ni trop aérée, et que des lignes consécutives aient à peu près la même densité.

Vous pouvez cependant insérer l'instruction `\break` à l'endroit où vous le jugez utile pour « forcer » le passage à la ligne suivante :

```
c4 c c c | \break
c4 c c c |
```



Par défaut, un saut de ligne ne saurait intervenir au beau milieu d'une mesure ; LilyPond vous le signalera par un message si tel était le cas. Si d'aventure vous voulez forcer un saut de ligne en l'absence de barre de mesure, vous devrez auparavant insérer une barre invisible – à l'aide de `\bar ""`.

```
c4 c c
\bar "" \break
c |
c4 c c c |
```



LilyPond ignorera un `\break` placé sur une barre à la fin d'une mesure dès lors que la précédente avait une note en suspend – c'est typiquement le cas lorsqu'un nolet est à cheval sur deux mesures. L'instruction `\break` sera alors opérationnelle si vous avez auparavant désactivé le `Forbid_line_break_engraver` du contexte `Voice` concerné. Notez bien qu'en pareil cas, les sauts de ligne forcés doivent être saisis au sein d'une expression polyphonique :

```
\new Voice \with {
  \remove "Forbid_line_break_engraver"
} \relative c'' {
  <<
    { c2. \times 2/3 { c4 c c } c2. | }
    { s1 | \break s1 | }
```



```
>>
}
```



Selon le même principe, un saut de ligne ne peut intervenir alors qu'une ligature s'étend sur deux mesures consécutives. Il faut en ce cas là introduire la dérogation `\override Beam #'breakable = ##t`.

```
\override Beam #'breakable = ##t
c2. c8[ c | \break
c8 c] c2. |
```



L'instruction opposée, `\noBreak`, interdira toute tentative de saut de ligne à la fin de la mesure où elle est explicitée.

LilyPond dispose de deux variables de base pour influencer l'espacement au niveau des lignes. Toutes deux se définissent dans un bloc `\layout`, `indent` réglant l'indentation de la première ligne, et `line-width` la longueur des lignes.

L'activation du commutateur `ragged-right` au sein du bloc `\layout` aura pour effet de terminer les systèmes là où ils prendraient fin normalement plutôt que de les étirer sur toute la longueur de la ligne. Ceci est particulièrement utile pour de petits fragments ou pour vérifier la densité induite par l'espacement naturel.

Le commutateur `ragged-last` est équivalent à `ragged-right`, à ceci près qu'il n'affecte que la dernière ligne de la pièce.

```
\layout {
  indent = 0\mm
  line-width = 150\mm
  ragged-last = ##t
}
```

L'utilisation conjointe de `\break` et de blancs dans une section `\repeat` vous permettra de positionner des sauts de ligne à intervalle régulier. Par exemple, les 28 mesures de ce qui suit, si l'on est à 4/4, seront coupées toutes les quatre mesures, pas ailleurs :

```
<<
\repeat unfold 7 {
  s1 \noBreak s1 \noBreak
  s1 \noBreak s1 \break
}
```

```

    }
    { et ici la musique... }
>>

```

Commandes prédéfinies

`\break`, `\noBreak`.

Voir aussi

Manuel de notation : [\[Variables de gestion des sauts de ligne\]](#), page 504.

Référence des propriétés internes : [Section “LineBreakEvent” dans Référence des propriétés internes](#).

Morceaux choisis : [Section “Espaces” dans Morceaux choisis](#).

4.3.2 Sauts de page

La gestion automatique des sauts de page se contrôle à l'aide des commandes `\pageBreak` et `\noPageBreak`. Ces commandes fonctionnent de manière analogue à `\break` et `\noBreak` pour les sauts de ligne et se placent donc au moment d'une barre de mesure. Elles permettent de forcer, ou d'interdire, un saut de page à l'endroit indiqué. Comme on peut s'y attendre, `\pageBreak` force le saut de ligne.

Les commandes `\pageBreak` et `\noPageBreak` peuvent se trouver à des niveaux supérieurs, entre deux partitions ou *markups* de premier rang.

Tout comme `ragged-right` et `ragged-last` qui permettent de gérer la répartition horizontale, LilyPond dispose de commutateurs équivalents au niveau de la verticalité : `ragged-bottom` et `ragged-last-bottom`. Lorsqu'ils sont tous deux activés – affectés de `#t` – les systèmes ne seront pas répartis sur les pages y compris la dernière. Pour de plus amples détails, reportez-vous à [Section 4.1.3 \[Variables d'espacement vertical fixe\]](#), page 499.

Les sauts de page sont générés par la fonction `page-breaking`. LilyPond dispose de trois différents algorithmes en la matière : `ly:optimal-breaking`, `ly:page-turn-breaking` et `ly:minimal-breaking`. C'est `ly:optimal-breaking` qui est activé par défaut, mais rien ne vous empêche d'en changer, par l'intermédiaire du bloc `\paper` :

```

\paper {
  page-breaking = #ly:page-turn-breaking
}

```

Lorsqu'un ouvrage contient plusieurs partitions et un certain nombre de pages, la gestion des sauts de page finit par devenir très gourmande, tant au niveau du processeur que de la mémoire. Vous pouvez cependant alléger la charge en recourant à des blocs `\bookpart` afin de sectionner l'ouvrage que vous traitez ; les sauts de page seront alors gérés individuellement au niveau de chacune des parties. Par ailleurs, cela vous autorisera une gestion différente selon les sections.

```

\bookpart {
  \header {
    subtitle = "Préface"
  }
  \paper {
    %% Pour une partie constituée principalement de texte
    %% ly:minimal-breaking est plus judicieux.
    page-breaking = #ly:minimal-breaking
  }
  \markup { ... }
  ...
}

```

```
\bookpart {
  %% Cette partie étant purement musicale,
  %% retour au style par défaut (optimal-breaking).
  \header {
    subtitle = "Premier mouvement"
  }
  \score { ... }
  ...
}
```

Commandes prédéfinies

`\pageBreak`, `\noPageBreak`.

Voir aussi

Manuel de notation : [\[Variables de gestion des sauts de page\]](#), page 504.

Morceaux choisis : [Section “E spacements”](#) dans *Morceaux choisis*.

4.3.3 Optimisation des sauts de page

LilyPond, pour déterminer où placer un saut de page, utilise par défaut la fonction `ly:optimal-breaking`. Celle-ci tend à trouver une rupture qui évite d’obtenir à la fois une page trop dense ou exagérément aérée. Contrairement à la fonction `ly:page-turn-breaking`, elle n’a aucune notion de ce qu’est une « tourne ».

Voir aussi

Morceaux choisis : [Section “E spacements”](#) dans *Morceaux choisis*.

4.3.4 Optimisation des tournes

Aboutir à une configuration des sauts de page de telle sorte que les pages de droite se terminent toujours par un silence devient souvent une nécessité. En effet, l’exécutant pourra alors tourner la page sans risquer de manquer des notes. La fonction `ly:page-turn-breaking` tend à trouver une rupture qui évite d’obtenir à la fois une page trop dense ou exagérément aérée, tout en tenant compte du fait qu’une tourne ne saurait intervenir qu’à certains endroits.

L’utilisation de cette fonction se fait en deux étapes. Il vous faut tout d’abord l’activer au sein du bloc `\paper` comme indiqué à la rubrique [Section 4.3.2 \[Sauts de page\]](#), page 512. Vous devrez, dans un deuxième temps, informer la fonction des endroits où les sauts de page sont permis.

Cette deuxième étape se réalise de deux manières différentes. Vous pouvez spécifier manuellement chaque tourne potentielle en insérant un `\allowPageTurn` à l’endroit approprié de votre fichier source.

Toutefois, cette option peut vite se révéler fastidieuse selon l’ampleur de l’œuvre. Vous pouvez alors recourir au `Page_turn_engraver` que vous mentionnerez dans un contexte de voix ou de portée. Ce graveur de tournes recherchera dans le contexte en question les passages sans note. Notez bien qu’il ne recherche pas des silences, mais l’absence de notes ; autrement dit, il ne restera pas inactif dans le cadre d’une portée polyphonique dont l’une des parties contiendrait des silences. Lorsqu’il rencontre un fragment suffisamment long ne contenant aucune note, il insère un `\allowPageTurn` à la barre terminant ce fragment, à moins qu’il ne rencontre en chemin une « barre spéciale » – telle une double barre – auquel cas il y déposera le `\allowPageTurn`.

Le `Page_turn_engraver` examine la propriété de contexte `minimumPageTurnLength` pour déterminer quelle doit être la longueur d’un fragment sans note avant une tourne. La valeur par défaut de `minimumPageTurnLength` est (`ly:make-moment 1 1`), soit une ronde, et s’ajuste de la manière suivante :

```

\new Staff \with { \consists "Page_turn_engraver" }
{
  a4 b c d |
  R1 | % une tourne peut se placer ici
  a4 b c d |
  \set Staff.minimumPageTurnLength = #(ly:make-moment 5 2)
  R1 | % il ne peut pas y avoir de tourne ici
  a4 b r2 |
  R1*2 | % une tourne peut se placer ici
  a1
}

```

Le `Page_turn_engraver` tient compte des reprises. C'est pourquoi il ne permettra une tourne que dans le cas où il y aura suffisamment de temps au début et à la fin de la reprise pour que l'exécutant ait le temps de revenir à la page précédente. Le `Page_turn_engraver` est même capable d'interdire un tourne dans le cas d'une reprise de courte durée, ajustable au travers de la propriété de contexte `minimumRepeatLengthForPageTurn`.

Les commandes de tourne – `\pageTurn`, `\noPageTurn` et `\allowPageTurn` – peuvent s'utiliser à des niveaux supérieurs, entre des blocs `\score` ou des *markups* de haut niveau.

Commandes prédéfinies

`\pageTurn`, `\noPageTurn`, `\allowPageTurn`.

Voir aussi

Morceaux choisis : [Section “E spacements” dans *Morceaux choisis*](#).

Problèmes connus et avertissements

Une partition ne devrait contenir qu'une seule instance du `Page_turn_engraver`, au risque de les voir se contredire.

4.3.5 Minimisation des sauts de page

La fonction `ly:minimal-breaking` est celle qui réalise le moins de calculs pour positionner les sauts de page. Elle mettra le plus de systèmes possible sur une page avant de passer à la suivante. On peut donc la préférer lorsque la partition s'étend sur beaucoup de pages ou lorsque les autres fonctions de gestion des sauts de page ralentissent nettement le traitement, sont trop gourmandes en mémoire ou qu'il y a beaucoup de texte. Il suffit de la mentionner au sein du bloc `\paper` :

```

\paper {
  page-breaking = #ly:minimal-breaking
}

```

Voir aussi

Morceaux choisis : [Section “E spacements” dans *Morceaux choisis*](#).

4.3.6 Présentation en rouleau

La fonction `ly:one-line-breaking` constitue un algorithme de calcul des sauts de pages particulier en ceci que chaque partition fait l'objet d'une page unique, d'une seule ligne. Cette fonctionnalité s'affranchit de l'impression des titres et marges ; seule la partition est affichée.

La largeur de page est ajustée de telle sorte que la pièce la plus longue tienne sur une seule ligne. En particulier, les variables `paper-width`, `line-width` et `indent` du bloc `\paper` seront ignorées ; les `left-margin` et `right-margin` seront honorées. La hauteur de page ne sera pas modifiée.

4.3.7 Sauts explicites

Il arrive parfois que LilyPond rejette des `\break` ou des `\pageBreak` explicites. Vous pouvez alors prendre le contrôle avec ces deux instructions dérogatoires :

```
\override NonMusicalPaperColumn #'line-break-permission = ##f
\override NonMusicalPaperColumn #'page-break-permission = ##f
```

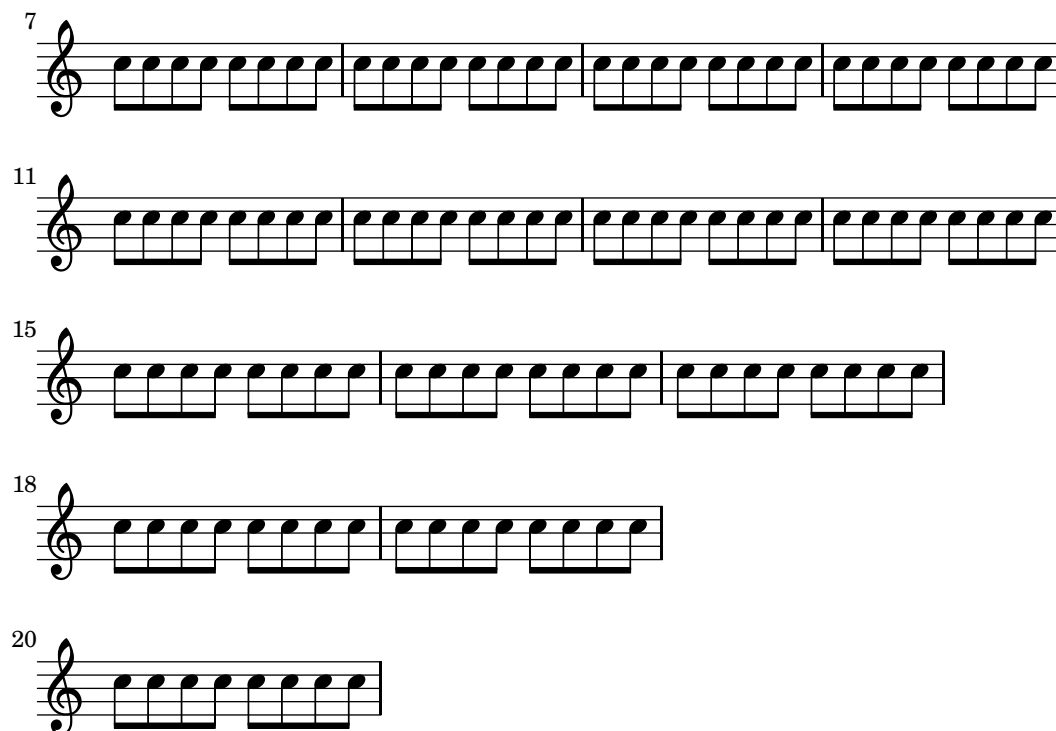
Lorsque vous désactivez `line-break-permission`, LilyPond ne passera à la ligne suivante qu'en présence d'un `\break` explicite, et nulle part ailleurs. De la même façon, la désactivation de `page-break-permission` aura pour conséquence que LilyPond ne changera de page que lorsqu'il rencontrera un `\pageBreak`, et nulle part ailleurs.

```
\paper {
  indent = #0
  ragged-right = ##t
  ragged-bottom = ##t
}
```

```
musique = \relative c'' { c8 c c c }
```

```
\score {
  \new Staff {
    \repeat unfold 2 { \musique } \break
    \repeat unfold 4 { \musique } \break
    \repeat unfold 6 { \musique } \break
    \repeat unfold 8 { \musique } \pageBreak
    \repeat unfold 8 { \musique } \break
    \repeat unfold 6 { \musique } \break
    \repeat unfold 4 { \musique } \break
    \repeat unfold 2 { \musique }
  }
  \layout {
    \context {
      \Score
      \override NonMusicalPaperColumn #'line-break-permission = ##f
      \override NonMusicalPaperColumn #'page-break-permission = ##f
    }
  }
}
```





Voir aussi

Morceaux choisis : [Section “E spacements”](#) dans *Morceaux choisis*.

4.3.8 Recours à une voix supplémentaire pour gérer les sauts

La plupart du temps, les informations concernant les sauts de ligne ou de page se retrouvent directement au milieu des notes.

```
musique = \relative c'' { c4 c c c }
```

```
\score {
  \new Staff {
    \repeat unfold 2 { \musique } \break
    \repeat unfold 3 { \musique }
  }
}
```

Bien que cela constitue un moyen aisé de saisir les commandes `\break` et `\pageBreak`, les données musicales se retrouvent mélangées à des informations qui concernent plutôt l’agencement de la musique sur le papier. Vous pouvez tout à fait séparer ce qui est purement musical et les informations concernant les sauts de ligne ou de page en créant une voix supplémentaire dédiée. Cette voix spécifique ne contiendra que des blancs – des silences invisibles `\skip` –, des `\break`, des `\pageBreak` et autres informations concernant les ruptures.

```
music = \relative c'' { c4 c c c }
```

```
\score {
  \new Staff <<
    \new Voice {
      s1 * 2 \break
      s1 * 3 \break
      s1 * 6 \break
      s1 * 5 \break
    }
    \new Voice {
```

```

\repeat unfold 2 { \music }
\repeat unfold 3 { \music }
\repeat unfold 6 { \music }
\repeat unfold 5 { \music }
}
>>
}

```



Cette manière de procéder est tout à fait indiquée lorsque vous ajustez les `line-break-system-details` et autres propriétés fort intéressantes de `NonMusicalPaperColumnGrob`, comme vous pouvez le voir au chapitre [Section 4.4 \[Espace vertical\]](#), page 518.

```
music = \relative c'' { c4 c c c }
```

```

\score {
  \new Staff <<
    \new Voice {
      \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 0))
      s1 * 2 \break

      \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 35))
      s1 * 3 \break

      \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 70))
      s1 * 6 \break

      \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 105))
      s1 * 5 \break
    }
  \new Voice {
    \repeat unfold 2 { \music }
    \repeat unfold 3 { \music }
  }
}

```

```

\repeat unfold 6 { \music }
\repeat unfold 5 { \music }
}
>>
}

```



Voir aussi

Manuel de notation : [Section 4.4 \[Espace vertical\]](#), page 518.

Morceaux choisis : [Section “Espace vertical”](#) dans *Morceaux choisis*.

4.4 Espace vertical

L'espace vertical dépend de trois éléments : la surface disponible – par exemple format de papier et marges –, l'espace qui doit séparer les systèmes, et l'espace qui sépare les portées d'un même système.

4.4.1 Espace vertical au sein d'un système

LilyPond dispose de trois différents mécanismes permettant de contrôler l'espace au sein d'un système selon trois catégories :

- *portées isolées*,
- *portées regroupées* (portées d'un même groupe, telles celles d'un `ChoirStaff`, etc.), et
- *lignes de non-portée* (`Lyrics`, `ChordNames`, etc.).

La hauteur de chaque système se détermine en deux phases. Les portées sont tout d'abord espacées selon la surface disponible. Puis les lignes autres que des portées, comme les paroles ou les accords, sont réparties entre les portées.

Les paragraphes qui suivent traitent exclusivement de la manière de gérer l'espace entre les lignes d'un système – portée musicale ou non. Pour ce qui a trait aux espacements entre les systèmes, mouvements, annotations et marge, ils sont contrôlés par des variables attachées au bloc `\paper` et font l'objet du chapitre [Section 4.1.4 \[Variables d'espace vertical fluctuant\]](#), page 500.

Propriétés d'espacement au sein d'un système

L'espacement entre les portées est géré par deux jeux de propriétés d'objet graphique (*grob*). Le premier, associé à l'objet graphique **VerticalAxisGroup**, est créé pour toute ligne de portée ou de non-portée. Le second, associé à l'objet graphique **StaffGrouper**, doit être explicitement créé pour un regroupement de portées particulier. Les propriétés qui leur sont attachées sont abordées en fin de section.

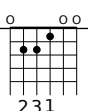
Le nom de ces propriétés, sauf **staff-affinity**, suit le schéma *item1-item2-spacing* – *item1* et *item2* étant les éléments à espacer. Notez bien que *item2* n'est pas forcément placé au-dessous : c'est le cas pour la propriété **nonstaff-relatedstaff-spacing** qui spécifie l'espacement d'une ligne de non-portée alors que sa **staff-affinity** a été déterminée à UP.

Toutes ces distances sont mesurées entre les points de référence respectifs des éléments considérés. Le *point de référence* d'une portée est le centre vertical du **StaffSymbol** – la ligne médiane si **line-count** est impair, l'interligne médian si **line-count** est pair. Quant aux lignes rattachées à des portées – lignes de non-portée – le tableau suivant présente le *point de référence* pour chacune d'elles :

Ligne de non-portée	Point de référence
ChordNames	ligne de base
NoteNames	ligne de base
Lyrics	ligne de base
Dynamics	centre vertical
FiguredBass	point le plus haut
FretBoards	ligne supérieure

En voici une représentation graphique :

Diagram illustrating the reference points for various musical notation elements on a staff:

- ChordNames**: ligne de base (baseline) — *g*
- NoteNames**: ligne de base — *g*
- Lyrics**: ligne de base — *ghijk*
- Dynamics**: centre vertical — *mp* — *fp*
- FiguredBass**: point le plus haut — *6*/*5*
- FretBoards**: ligne du haut — 

Hormis **staff-affinity** – propriété attachée au *grob* **VerticalAxisGroup** –, chacune de ces propriétés est enregistrée sous la forme d'une liste associative dont la structure est identique à celle des variables du bloc **\paper** que nous avons examinées au chapitre [Section 4.1.4 \[Variables d'espacement vertical fluctuant\]](#), page 500. Les particularités en matière de modification d'une liste associative font l'objet d'un [Section "chapitre particulier" dans Manuel de notation](#). Les propriétés des objets graphiques se règlent avec un **\override** mentionné dans un bloc **\score** ou **\layout**, pas dans le bloc **\paper**.

L'exemple suivant illustre deux façons de modifier une liste associative. La première déclaration n'agit que sur une seule clé, alors que la seconde redéfinit la propriété dans son intégralité.

```
\new Staff \with {
  \override VerticalAxisGroup #'staff-staff-spacing #'basic-distance = #10
} { ... }
```

```
\new Staff \with {
  \override VerticalAxisGroup #'staff-staff-spacing =
    #'(('basic-distance . 10)
      (minimum-distance . 9)
      (padding . 1)
      (stretchability . 10))
} { ... }
```

La modification d'un espacement au niveau global se mentionne au sein du bloc `\layout` :

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup #'staff-staff-spacing #'basic-distance = #10
  }
}
```

Les réglages concernant les propriétés d'espacement vertical des objets graphiques sont répertoriées aux chapitres [Section “VerticalAxisGroup” dans Référence des propriétés internes](#) et [Section “StaffGrouper” dans Référence des propriétés internes](#). Les propriétés relatives aux lignes de non-portée sont répertoriées selon la définition de leur contexte dans la [Section “Contexts” dans Référence des propriétés internes](#).

Propriétés de l'objet VerticalAxisGroup

Les propriétés de l'objet `VerticalAxisGroup` s'ajustent à l'aide d'un `\override` au niveau d'un contexte `Staff` (ou son équivalent).

staff-staff-spacing

Il s'agit de la distance entre la portée en cours et la portée qui suit au sein du même regroupement, qu'il y ait ou non une ligne de non-portée (`Lyrics` ou autre) entre les deux. Cette propriété ne s'applique pas à la dernière portée d'un système.

En tout état de cause, la fonction Scheme `staff-staff-spacing` d'un `VerticalAxisGroup` affectera les propriétés du `StaffGrouper` si la portée est incluse dans un regroupement ; elle s'appliquera au `default-staff-staff-spacing` en l'absence de regroupement. Les portées peuvent donc s'aligner différemment selon qu'elles sont ou non regroupées. Pour obtenir le même espacement sans tenir compte des éventuels regroupements, cette fonction peut faire place à une complète redéfinition des espacements fluctuants à l'aide de règles dérogatoires comme vu précédemment.

default-staff-staff-spacing

Il s'agit de la distance qui s'appliquera par défaut aux portées isolées, à moins que `staff-staff-spacing` n'ait été redéfini explicitement par un `\override`.

staff-affinity

Il s'agit de la direction – `UP`, `DOWN` ou `CENTER` – que prendra une ligne de non-portée pour aller s'accoler aux portées adjacentes . Si vous lui attribuez `CENTER`, cette ligne de non-portée ira se placer à équidistance entre les portées qui l'encadrent,

tout en tenant compte des éventuels risques de collision et des autres contraintes d'espacement. Des lignes de non-portée adjacentes devraient avoir une **staff-affinity** allant de haut en bas – autrement dit, pas de UP après un DOWN. Une ligne de non-portée en dessous d'un système devrait avoir sa **staff-affinity** définie à UP. De la même manière, lorsque cette ligne surplombe un système, sa **staff-affinity** devrait être définie à DOWN. Prenez garde à la valeur que vous affectez à **staff-affinity** : si vous affectez la valeur **#f** à une ligne de non-portée, cette ligne sera considérée comme étant une portée ; à l'inverse, utiliser la propriété **staff-affinity** pour une portée lui fera perdre cette qualité.

nonstaff-relatedstaff-spacing

Il s'agit de la distance entre la ligne de non-portée en cours et la portée la plus proche selon la **staff-affinity**, à la double condition qu'il n'y ait pas déjà une autre ligne de non-portée et que la valeur de **staff-affinity** soit UP ou DOWN. Lorsque la valeur de **staff-affinity** est égale à **CENTER**, la valeur de **nonstaff-relatedstaff-spacing** servira à centrer la ligne de non-portée entre les deux portées adjacentes même si une autre non-portée est présente (quelque soit le côté).

nonstaff-nonstaff-spacing

Il s'agit de la distance entre deux lignes de non-portée selon l'orientation définie par **staff-affinity** et dès lors qu'elles ont la même orientation. Bien entendu, ceci ne peut concerner que les valeurs UP et DOWN de **staff-affinity**.

nonstaff-unrelatedstaff-spacing

Il s'agit de la distance entre une ligne de non-portée et la portée à l'opposé de l'orientation adoptée, à la double condition qu'il n'y ait pas déjà une autre ligne de non-portée et que la valeur de **staff-affinity** soit UP ou DOWN. Cette propriété trouve toute sa légitimité pour décaler une ligne de **Lyrics** de la portée à laquelle elle ne correspond pas.

Propriétés de l'objet StaffGrouper

Les propriétés de l'objet **StaffGrouper** s'ajustent à l'aide d'un **\override** au niveau d'un contexte **StaffGroup** (ou son équivalent).

staff-staff-spacing

Il s'agit de la distance entre deux portées consécutives d'un même système. La propriété **staff-staff-spacing** de l'objet **VerticalAxisGroup** d'une portée en particulier peut se redéfinir à l'aide de règles déroatoires.

staffgroup-staff-spacing

Il s'agit de la distance entre la dernière portée d'un regroupement et la portée suivante, au sein d'un même système, y compris lorsqu'une ou plusieurs lignes de non-portée (tel **Lyrics**) s'insèrent entre les deux. Cette propriété ne concerne pas la dernière portée d'un système. Dans le cas où la propriété **staff-staff-spacing** d'une portée du regroupement a été ajustée au niveau de son propre **VerticalAxisGroup**, cette dernière aura préséance.

Voir aussi

Fichiers d'initialisation : **'ly/engraver-init.ly'**, **'scm/define-grobs.scm'**.

Référence des propriétés internes : Section “Contexts” dans *Référence des propriétés internes*, Section “VerticalAxisGroup” dans *Référence des propriétés internes*, Section “StaffGrouper” dans *Référence des propriétés internes*.

Espacement de portées isolées

Les `Staff`, `DrumStaff`, `TabStaff` entre autres sont des contextes de « portée » pouvant contenir plusieurs voix, mais pas une portée.

L'espacement de ces *portées isolées* est géré par les propriétés suivantes :

- Propriétés du `VerticalAxisGroup` :
 - `default-staff-staff-spacing`
 - `staff-staff-spacing`

Ces propriétés d'objet graphique sont expliquées une à une au chapitre [\[Propriétés d'espacement au sein d'un système\]](#), page 519.

Certaines propriétés supplémentaires s'appliqueront dès lors que ces portées sont regroupées – voir [\[Espacement de portées regroupées\]](#), page 523.

L'exemple suivant illustre la manière de gérer l'espacement de portées isolées à l'aide de la propriété `default-staff-staff-spacing`. Les mêmes règles appliquées de manière dérogatoire au `staff-staff-spacing` produiront les mêmes effets, y compris au sein de regroupements.

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup #'default-staff-staff-spacing =
      #'((basic-distance . 8)
        (minimum-distance . 7)
        (padding . 1))
  }
}

<<
% The very low note here needs more room than 'basic-distance
% can provide, so the distance between this staff and the next
% is determined by 'padding.
\new Staff { b,2 r | }

% Here, 'basic-distance provides enough room, and there is no
% need to compress the space (towards 'minimum-distance) to make
% room for anything else on the page, so the distance between
% this staff and the next is determined by 'basic-distance.
\new Staff { \clef bass g2 r | }

% By setting 'padding to a negative value, staves can be made to
% collide. The lowest acceptable value for 'basic-distance is 0.
\new Staff \with {
  \override VerticalAxisGroup #'default-staff-staff-spacing =
    #'((basic-distance . 3.5)
      (padding . -10))
} { \clef bass g2 r | }
\new Staff { \clef bass g2 r | }
>>
```



Voir aussi

Fichiers d'initialisation : `'scm/define-grobs.scm'`.

Morceaux choisis : [Section "E spacements"](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section "VerticalAxisGroup"](#) dans *Référence des propriétés internes*.

E spacement de portées regroupées

Dans les partitions orchestrales ou de grande ampleur, il arrive souvent que des portées soient regroupées. L'espace est alors plus important entre deux regroupements qu'entre les portées d'un même groupe.

Les *regroupements de portées* tels le `StaffGroup` ou le `ChoirStaff` sont des contextes qui peuvent contenir simultanément une ou plusieurs portées.

L'espace entre les portées d'un même regroupement est géré par les propriétés suivantes :

- Propriétés du `VerticalAxisGroup` :
 - `staff-staff-spacing`
- Propriétés du `StaffGrouper` :
 - `staff-staff-spacing`
 - `staffgroup-staff-spacing`

Ces propriétés d'objet graphique sont expliquées une à une au chapitre [\[Propriétés d'espace au sein d'un système\]](#), page 519.

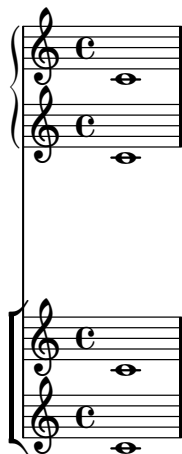
L'exemple suivant illustre la manière de gérer l'espace de portées regroupées, à l'aide des propriétés de l'objet graphique `StaffGrouper` :

```
\layout {
  \context {
    \Score
    \override StaffGrouper #'staff-staff-spacing #'padding = #0
    \override StaffGrouper #'staff-staff-spacing #'basic-distance = #1
  }
}

<<
\new PianoStaff \with {
  \override StaffGrouper #'staffgroup-staff-spacing #'basic-distance = #20
} <<
  \new Staff { c'1 }
  \new Staff { c'1 }
>>

\new StaffGroup <<
  \new Staff { c'1 }
  \new Staff { c'1 }
```

```
>>
>>
```



Voir aussi

Fichiers d'initialisation : `'scm/define-grobs.scm'`.

Morceaux choisis : [Section "E spacements"](#) dans *Morceaux choisis*.

Référence des propriétés internes: [Section "VerticalAxisGroup"](#) dans *Référence des propriétés internes*, [Section "StaffGrouper"](#) dans *Référence des propriétés internes*.

E spacement des lignes rattachées à des portées

Les *lignes de non-portée*, comme les `Lyrics` ou les `ChordNames` sont des contextes dont les objets de rendu sont gravés à l'instar des portées – une ligne horizontale dans un système. En fait, les lignes de non-portée sont des contextes qui vont créer un objet de rendu `VerticalAxisGroup`.

L'espace ment des lignes de non-portée est géré par les propriétés suivantes :

- Propriétés du `VerticalAxisGroup` :
 - `staff-affinity`
 - `nonstaff-relatedstaff-spacing`
 - `nonstaff-nonstaff-spacing`
 - `nonstaff-unrelatedstaff-spacing`

Ces propriétés d'objet graphique sont expliquées une à une au chapitre [\[Propriétés d'espace ment au sein d'un système\]](#), page 519.

L'exemple suivant utilise la propriété `nonstaff-nonstaff-spacing` pour gérer l'espace ment entre des lignes consécutives de non-portée. Vous noterez que la valeur élevée attribuée à la clé `stretchability` permet aux paroles de s'étirer plus que de raison.

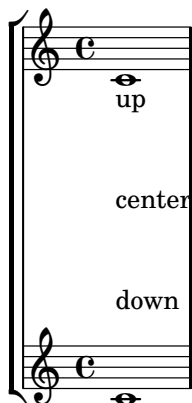
```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup
      #'nonstaff-nonstaff-spacing #'stretchability = #1000
  }
}

\new StaffGroup
<<
```

```

\new Staff \with {
  \override VerticalAxisGroup #'staff-staff-spacing = #'((basic-distance . 30))
} { c'1 }
\new Lyrics \with {
  \override VerticalAxisGroup #'staff-affinity = #UP
} \lyricmode { up }
\new Lyrics \with {
  \override VerticalAxisGroup #'staff-affinity = #CENTER
} \lyricmode { center }
\new Lyrics \with {
  \override VerticalAxisGroup #'staff-affinity = #DOWN
} \lyricmode { down }
\new Staff { c'1 }
>>

```



Voir aussi

Fichiers d'initialisation : 'ly/engraver-init.ly', 'scm/define-grobs.scm'.

Morceaux choisis : [Section "Espacements"](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section "Contexts"](#) dans *Référence des propriétés internes*, [Section "VerticalAxisGroup"](#) dans *Référence des propriétés internes*.

4.4.2 Positionnement explicite des portées et systèmes

Pour bien comprendre comment fonctionnent les réglages de `VerticalAxisGroup` et de `\paper` abordés dans les deux rubriques précédentes, rien ne vaut une collection d'exemples illustrant les différentes mises au point du décalage vertical appliqué aux portées et systèmes distribués sur une page.

Une autre approche de l'espacement vertical est le recours à `NonMusicalPaperColumn` `#'line-break-system-details`. Alors que `VerticalAxisGroup` et `\paper` gèrent un décalage vertical, `NonMusicalPaperColumn` `#'line-break-system-details` spécifiera le positionnement vertical absolu sur la page.

`NonMusicalPaperColumn` `#'line-break-system-details` prend en charge une liste associative de trois mises au point :

- `X-offset`
- `Y-offset`
- `alignment-distances`

Les dérogations en matière d'objet graphique, y compris celles concernant les `NonMusicalPaperColumn` ci-dessus, peuvent se placer à trois différents endroits de votre fichier source :

- directement au beau milieu des notes
- au sein d'un bloc `\context`
- dans un bloc `\with`

Le réglage de `NonMusicalPaperColumn` s'effectue à l'aide d'une simple commande `\override` au sein d'un bloc `\context` ou `\with`. Dans le cas où il est stipulé au fil des notes, c'est la commande spécifique `\overrideProperty` qui doit intervenir. Voici quelques exemples de réglages de `NonMusicalPaperColumn` à l'aide de la commande `\overrideProperty` :

```
\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((X-offset . 20))

\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((Y-offset . 40))

\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((X-offset . 20)
                                (Y-offset . 40))

\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((alignment-distances . (15)))

\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((X-offset . 20)
                                (Y-offset . 40)
                                (alignment-distances . (15)))
```

Nous allons maintenant voir ces différents réglages en action. Commençons par examiner un exemple dépourvu de toute mise au point.

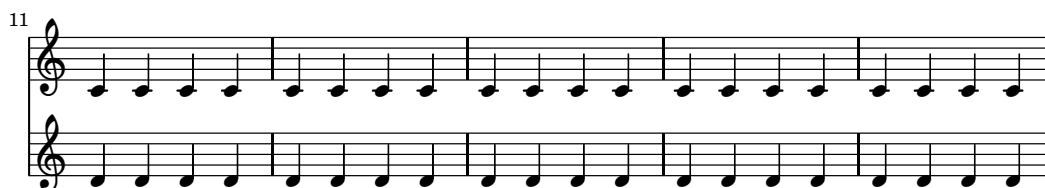
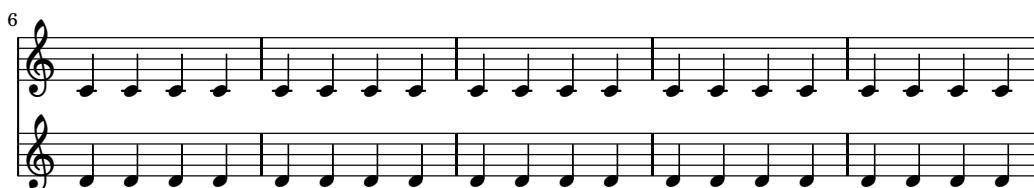
```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
    \new Staff <<
    \new Voice {
      s1*5 \break
      s1*5 \break
      s1*5 \break
    }
    \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
    >>
    \new Staff {
      \repeat unfold 15 { d'4 d' d' d' }
    }
    >>
  }
}
```




Cette partition isole les informations de saut de ligne ou de page dans une voix spécifique. La mise en forme est ainsi séparée des événements musicaux ; ceci nous permettra d'y voir plus clair au fur et à mesure que nous avancerons. Pour plus de précisions, relisez [Section 4.3.8 \[Recours à une voix supplémentaire pour gérer les sauts\]](#), page 516.

Les `\break` explicites répartissent la musique en lignes de cinq mesures chacune. L'espacement vertical est celui que LilyPond attribue par défaut. Nous pouvons, afin de fixer explicitement le point d'attache vertical de chacun des systèmes, définir un doublet `Y-offset` en tant qu'attribut du `line-break-system-details` de l'objet `NonMusicalPaperColumn` :

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 0))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 40))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 80))
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```



Vous aurez remarqué que nous n'avons déterminé qu'une seule valeur, même si la liste associative de `line-break-system-details` peut en comporter un certain nombre. Vous aurez aussi noté que la propriété `Y-offset` détermine ici le point de départ de chacun des systèmes de la page.

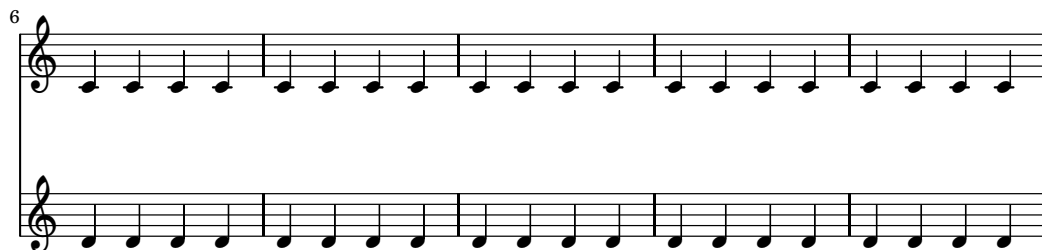
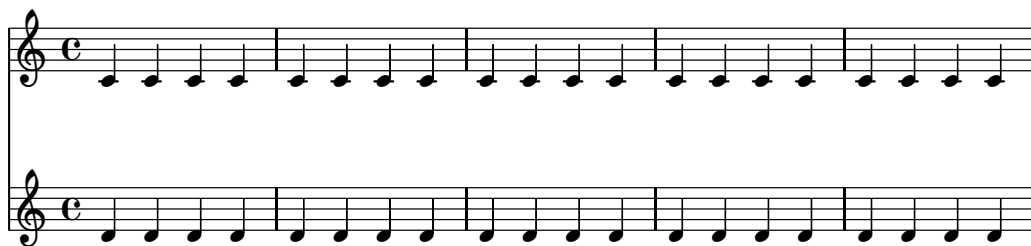
Maintenant que chaque système est explicitement positionné, nous pouvons jouer sur la distance séparant les portées de chacun des systèmes, grâce à la sous-propriété `alignment-distances` de `line-break-system-details`.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 20)
              (alignment-distances . (15)))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 60)
              (alignment-distances . (15)))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 100)
              (alignment-distances . (15)))
        }
      }
    }
  }
```

```

                                (alignment-distances . (15)))
      s1*5 \break
    }
    \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
  >>
  \new Staff {
    \repeat unfold 15 { d'4 d' d' d' }
  }
  >>
}
}

```



Nous avons maintenant assigné deux valeurs différentes à l'attribut `line-break-system-details` de l'objet `NonMusicalPaperColumn`. `line-break-system-details` pourrait prendre

bien d'autres paramètres d'espacement, y compris un doublet **X-offset**, mais nous n'avons utilisé que **Y-offset** et **alignment-distances** pour contrôler le positionnement de chaque système et de chaque portée. Vous noterez enfin que **alignment-distances** traite le positionnement des portées, non d'un regroupement de portées.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
    \new Staff <<
    \new Voice {
      \overrideProperty #"Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 0)
                                          (alignment-distances . (30 10)))

      s1*5 \break
      \overrideProperty #"Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 60)
                                          (alignment-distances . (10 10)))

      s1*5 \break
      \overrideProperty #"Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 100)
                                          (alignment-distances . (10 30)))

      s1*5 \break
    }
    \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
  >>
  \new StaffGroup <<
    \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
    \new Staff { \repeat unfold 15 { e'4 e' e' e' } }
  >>
  >>
}
}
```

The image displays three systems of musical notation, each consisting of three staves. The first system shows a single melodic line on the top staff, with the middle and bottom staves empty. The second system shows a piano accompaniment with a melody on the top staff and chords on the middle and bottom staves. The third system shows a similar piano accompaniment, but with the middle staff empty, leaving only the top and bottom staves with their respective parts. This illustrates how vertical alignment and spacing can be controlled in musical notation.

Quelques points à prendre en considération :

- Lorsque vous utilisez `alignment-distances`, les paroles et autres lignes de non-portée ne comptent pas pour une portée.
- Les nombres fournis à `X-offset`, `Y-offset` et `alignment-distances` sont considérés comme des multiples de la distance entre des portées adjacentes. Des valeurs positives remontent les portées et paroles, des valeurs négatives les descendent.
- Dans la mesure où `NonMusicalPaperColumn #'line-break-system-details` permet de positionner systèmes et portées n'importe où sur une page, vous pourriez être en contra-

diction avec les dimensionnements de la feuille ou bien aboutir à des surimpressions. Soyez donc raisonnable quant aux différentes valeurs que vous affectez à ces réglages.

Voir aussi

Morceaux choisis : [Section “E spacements” dans *Morceaux choisis*](#).

4.4.3 Résolution des collisions verticales

Vous savez de manière intuitive qu'un certain nombre d'objets en matière de notation musicale appartiennent à la portée, et que d'autres se placent en dehors de la portée. Entre autres objets externes, nous avons les marques repères, les textes et les nuances ; nous les appellerons « objets extérieurs à la portée ». La règle adoptée par LilyPond pour positionner verticalement ces objets extérieurs consiste à les placer au plus près de la portée tout en prenant garde d'éviter qu'il y ait chevauchement.

LilyPond utilise la propriété `outside-staff-priority` afin de déterminer si un objet est ou non un objet extérieur à la portée : lorsque la valeur de `outside-staff-priority` est numérique, il s'agit d'un objet extérieur à la portée. De plus, la propriété `outside-staff-priority` indique à LilyPond l'ordre dans lequel ces objets doivent être disposés.

Tout d'abord, LilyPond dispose tous les objets qui ne sont pas externes. Les objets extérieurs à la portée sont alors triés selon l'ordre croissant de leur `outside-staff-priority`. Enfin, LilyPond prend chacun des ces objets et les positionne de telle sorte qu'il n'entrent pas en collision avec ceux qui ont déjà été placés. Autrement dit, lorsque deux objets devraient se placer au même endroit, celui dont la `outside-staff-priority` est la plus faible sera disposé au plus près de la portée.

```
c4_"Text"\pp
r2.
\once \override TextScript #'outside-staff-priority = #1
c4_"Text"\pp % this time the text will be closer to the staff
r2.
% by setting outside-staff-priority to a non-number,
% we disable the automatic collision avoidance
\once \override TextScript #'outside-staff-priority = ##f
\once \override DynamicLineSpanner #'outside-staff-priority = ##f
c4_"Text"\pp % now they will collide
```

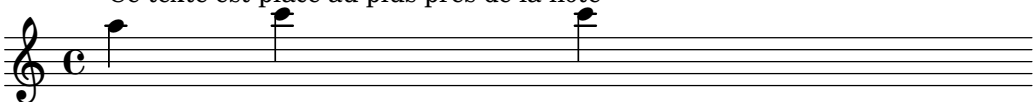


Le décalage vertical entre un objet extérieur à la portée et celui qui le précède se contrôle par la propriété `outside-staff-padding`.

```
\once \override TextScript #'outside-staff-padding = #0
a'~"Ce texte est placé au plus près de la note"
\once \override TextScript #'outside-staff-padding = #3
c~"Ce texte est décalé par rapport au texte précédent"
c~"Ce texte est accolé au texte précédent"
```

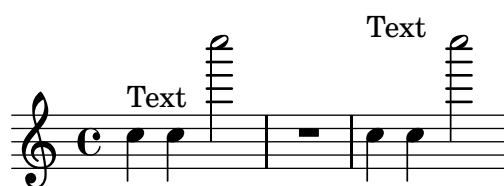
Ce texte est accolé au texte précédent
Ce texte est décalé par rapport au texte précédent

Ce texte est placé au plus près de la note



Par défaut, les objets extérieurs à la portée sont positionnés en évitant les collisions horizontales avec des objets précédemment positionnés. Ceci peut cependant générer des situations où des objets se trouvent horizontalement trop proches. L'espacement vertical entre les portées peut dans une certaine mesure permettre de mieux intercaler les objets extérieurs à la portée. La propriété `outside-staff-horizontal-padding` repoussera verticalement un objet pour éviter ce genre de situation.

```
% the markup is too close to the following note
c4^"Text"
c4
c''2
% setting outside-staff-horizontal-padding fixes this
R1
\once \override TextScript #'outside-staff-horizontal-padding = #1
c,,4^"Text"
c4
c''2
```



Voir aussi

Morceaux choisis : [Section “E spacements”](#) dans *Morceaux choisis*.

4.5 Espacement horizontal

4.5.1 Généralités sur l'espacement horizontal

Le moteur d'espacement traduit les différences de durée en distances étirables (*springs* pour ressorts) de différentes longueurs. Des durées importantes prennent ainsi plus de place que des durées moins longues. Les durées les plus courtes se verront attribuer un espace fixe, contrôlé par la propriété `shortest-duration-space` de l'objet [Section “SpacingSpanner”](#) dans *Référence des propriétés internes*. Au plus la durée s'allonge, au plus elle prendra d'espace : le doublement d'une durée attribuera à la note un espace fixé d'après la propriété `spacing-increment`.

L'exemple suivant comporte des blanches, des noires et un certain nombre de croches. La croche est suivie d'un espace de la largeur d'une tête de note ; pour la noire , cet espace est de deux têtes ; il est de trois pour la blanche.

```
c2 c4. c8 c4. c8 c4. c8 c8
c8 c4 c4 c4
```



spacing-increment est normalement défini à 1,2 espace de portée – ce qui correspond à peu près à la largeur d'une tête de note – et **shortest-duration-space** à 2,0. La note la plus courte s'étendra donc sur l'équivalent de 2,4 espaces de portée (2 fois le **spacing-increment**). Le point de départ de cet espace se situe à l'extrémité gauche du symbole ; la note la plus courte est donc suivie en général d'un espace égal à la largeur d'une tête de note.

Si l'on suit à la lettre ce qui précède, ajouter une simple triple croche à une partition qui comporte déjà des croches et des doubles augmentera considérablement son volume : la durée la plus courte n'est plus la double mais la triple croche, ce qui aura pour conséquence d'ajouter une largeur de tête à chacune des notes. Pour s'affranchir de cet effet quelque peu pervers, la durée la plus courte prise en considération au niveau de l'espacement n'est pas la note la plus brève de la partition, mais celle qui apparaît le plus souvent.

La courte durée la plus fréquente est déterminée à partir de la note la plus courte de chaque mesure. C'est elle qui servira de base pour l'espacement, à cette nuance près que la plus courte durée ne saurait être strictement supérieure à la croche. Cette « durée de référence » est d'ailleurs affichée lorsque vous lancez **lilypond** avec l'option '**--verbose**'.

Ces durées peuvent être adaptées. Vous pouvez définir la durée de base pour les espacements grâce à la propriété **common-shortest-duration** de l'objet **Section "SpacingSpanner"** dans *Référence des propriétés internes*. La durée maximale de cet étalon, normalement la croche, est gérée par la propriété **base-shortest-duration**.

Les notes plus courtes que la note témoin sont suivies d'un espace proportionnel à la durée témoin. Si donc nous ajoutions quelques doubles croches à l'exemple précédent, elles seraient suivies d'une demie largeur de tête :

c2 c4. c8 c4. c16[c] c4. c8 c8 c8 c4 c4 c4



Dans notre *Essai sur la gravure musicale automatisée*, nous avons vu comment la direction des hampes peut influencer l'espacement – voir **Section "Espacement"** dans *Essai*. Ceci est contrôlé par la propriété **stem-spacing-correction** de l'objet **Section "NoteSpacing"** dans *Référence des propriétés internes*, créé pour chaque contexte **Section "Voice"** dans *Référence des propriétés internes*. L'objet **Section "StaffSpacing"** dans *Référence des propriétés internes*, généré au niveau d'un contexte **Staff**, possède une même propriété qui contrôlera l'espacement hampe-barre de mesure. L'exemple suivant montre ces adaptations, tout d'abord selon les réglages par défaut, puis avec des corrections forcées.



L'espacement spécifique à la notation proportionnelle fait l'objet d'une **Section "rubrique dédiée"** dans *Manuel de notation*.

Voir aussi

Morceaux choisis : **Section "Espacements"** dans *Morceaux choisis*.

Essai sur la gravure musicale automatisée : **Section "Espacement"** dans *Essai*.

Référence des propriétés internes : **Section "SpacingSpanner"** dans *Référence des propriétés internes*, **Section "NoteSpacing"** dans *Référence des propriétés internes*, **Section "StaffSpacing"** dans *Référence des propriétés internes*, **Section "NonMusicalPaperColumn"** dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Il n'existe pas de mécanisme simple et efficace qui permette de forcer manuellement l'espacement. La solution ci-dessous permet cependant « d'aérer » artificiellement une partition ; il vous suffit d'ajuster la valeur du décalage (*padding*) autant que de besoin.

```
\override Score.NonMusicalPaperColumn #'padding = #10
```

Il n'y a aucun moyen de diminuer l'espacement.

4.5.2 Changement d'espacement en cours de partition

Il arrive, au cours d'un même mouvement, qu'une nouvelle partie modifie substantiellement la notion de valeur brève et valeur longue. La commande `newSpacingSection` permet alors de réinitialiser les paramètres d'espacement.

Dans l'exemple qui suit, le changement de métrique marque le début d'une nouvelle partie ; remarquez comme les doubles-croches sont alors plus espacées :

```
\time 2/4
c4 c8 c
c8 c c4 c16[ c c8] c4
\newSpacingSection
\time 4/16
c16[ c c8]
```



La commande `\newSpacingSection` crée un nouvel objet `SpacingSpanner`, ce qui permet d'appliquer de nouveaux `\overrides` à partir de ce point.

Voir aussi

Morceaux choisis : [Section “E spacements”](#) dans *Morceaux choisis*.

Référence des propriétés internes : [Section “SpacingSpanner”](#) dans *Référence des propriétés internes*.

4.5.3 Modification de l'espacement horizontal

Vous pouvez influencer l'espacement horizontal à l'aide de la propriété `base-shortest-duration`. Comparons les deux partitions qui suivent, toutes deux montrant la même musique. La première partition applique les réglages par défaut, alors que la seconde bénéficie d'un ajustement de la propriété `base-shortest-duration`. Au plus la valeur de `ly:make-moment` est grande, au plus la musique sera resserrée. En effet, `ly:make-moment` construit une durée : `1 4` est plus long que `1 16`.

```
\score {
  \relative c'' {
    g4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
}
```





```
\score {
  \relative c'' {
    g4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner
        #'base-shortest-duration = #(ly:make-moment 1 16)
    }
  }
}
```



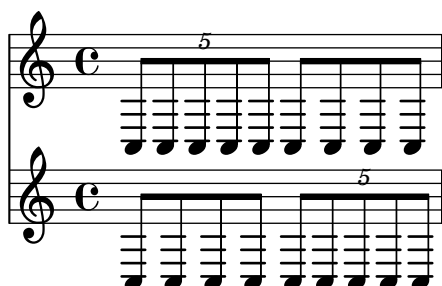
Morceaux choisis

L'espacement au sein d'un nolet dépend par défaut d'un certain nombre de facteurs qui ne sont pas liés à la durée (altération, changement de clef, etc.). `Score.SpacingSpanner #'uniform-stretching` permet d'ignorer ces symboles et, par voie de conséquence, forcer l'espacement sur la simple durée. Notez bien que cette propriété s'appliquera à toute la partition, puisque mentionnée au sein d'un bloc `\layout`.

```

\score {
  <<
    \new Staff {
      \times 4/5 {
        c8 c8 c8 c8 c8
      }
      c8 c8 c8 c8
    }
    \new Staff {
      c8 c8 c8 c8
      \times 4/5 {
        c8 c8 c8 c8 c8
      }
    }
  >>
  \layout {
    \context {
      \Score
      \override SpacingSpanner #'uniform-stretching = ##t
    }
  }
}

```

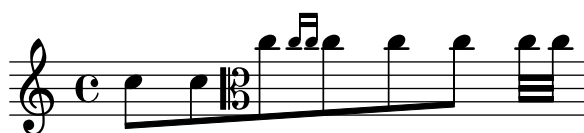


L'activation du commutateur `strict-note-spacing` permet d'espacer les notes sans tenir compte des clefs, barres de mesure ou notes d'ornement qui pourraient apparaître :

```

\override Score.SpacingSpanner #'strict-note-spacing = ##t
\new Staff { c8[ c \clef altoc \grace { c16[ c] } c8 c c] c32[ c32] }

```



Voir aussi

Morceaux choisis : [Section “E spacements”](#) dans *Morceaux choisis*.

4.5.4 Longueur de ligne

Deux réglages de bases ont une influence considérable sur l'espacement : `line-width` et `indent`. Tous deux se placent dans le bloc `\layout`. Ils contrôleront la longueur des lignes et l'indentation de la première.

L'activation du commutateur `ragged-right` au sein du bloc `\layout` permet de terminer les systèmes naturellement plutôt que de les voir s'étirer sur toute la largeur de la page. Cette

option est particulièrement utile lorsque vous traitez de courts fragments, ou bien pour vérifier ce que donnerait l'espacement naturel. Bien qu'il soit désactivé par défaut, il sera activé si la partition ne comporte qu'un seul système.

Le fonctionnement de l'option `ragged-last` est en tout point identique à celui de `ragged-right`, à ceci près qu'il ne concerne que la dernière ligne de la partition. Il n'y a pas de restriction quant à cette ligne. Il en va de même que pour le formatage d'un paragraphe de texte, la dernière ligne s'arrête au dernier caractère.

```
\layout {
  indent = #0
  line-width = #150
  ragged-last = ##t
}
```

Voir aussi

Morceaux choisis : [Section “E spacements” dans *Morceaux choisis*](#).

4.5.5 Notation proportionnelle

LilyPond prend en charge la notation proportionnelle. Il s'agit dans ce cas de représenter la notation selon un espacement strictement relatif aux durées. Ce type d'espacement pourrait se comparer à l'utilisation de papier millimétré pour positionner les notes au fil de la portée. Certaines œuvres de la fin du XXe siècle et à l'aube du XXIe utilisent cette proportionnalité dans le but de clarifier des structures rythmiques complexes, d'aider au positionnement d'indications temporelles ou autres éléments graphiques directement dans la partition.

LilyPond met à votre disposition cinq réglages différents, qui peuvent s'utiliser conjointement ou individuellement, aux fins de mettre au point cette notation proportionnelle.

- `proportionalNotationDuration`
- `uniform-stretching`
- `strict-note-spacing`
- `\remove "Separating_line_group_engraver"`
- `\override PaperColumn #'used = ##t`

Nous allons examiner, dans les différents exemples qui suivent, les effets de ces réglages et comment ils interagissent.

Commençons par cette mesure toute simple qui utilise l'espacement classique et justifiée à gauche.

```
\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
}
```

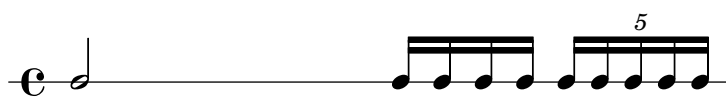


Vous constatez que la blanche qui entame la mesure prend moins de la moitié de l'espace. De même, les doubles croches et le quintolet de doubles (donc des vingtièmes de ronde) qui terminent cette mesure n'en occupent pas la moitié de l'espace horizontal.

En matière de gravure traditionnelle, cet espacement correspond tout à fait à nos attentes, puisque nous pouvons rogner l'espace sur la blanche et ainsi gagner en largeur sur toute la mesure qui fait une ronde.

Par contre, si nous avons besoin d'insérer une indication temporelle ou un autre graphisme en surplomb ou en dessous de notre partition, nous aurons besoin de la notation proportionnelle. Celle-ci s'active en définissant la propriété `proportionalNotationDuration`.

```
\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 20)
    }
  }
}
```



La blanche du début et les notes plus rapides de la deuxième moitié de la mesure occupent maintenant exactement le même espace horizontal. Nous pourrions donc y insérer, au-dessus ou au-dessous, une indication temporelle ou autre graphisme.

`proportionalNotationDuration` est une propriété attachée au contexte `Score`. Rappelez-vous que vous pouvez régler les propriétés d'un contexte à trois différents endroits de votre fichier : dans un bloc `\with`, dans un bloc `\context` ou au beau milieu de la musique à l'aide de la commande `\set`. Vous pouvez donc définir `proportionalNotationDuration` selon l'une de ces trois façons, à l'instar de n'importe quelle définition de contexte.

La propriété `proportionalNotationDuration` prend en unique argument la durée de référence qui servira de base pour espacer toute la musique. La fonction Scheme `make-moment` intégrée à LilyPond prend deux arguments : un numérateur et un dénominateur qui représentent une fraction de ronde. L'appel de `(ly:make-moment 1 20)` produit donc une durée de référence égale à un vingtième de ronde. Vous pourriez tout aussi bien utiliser `(ly:make-moment 1 16)`, `(ly:make-moment 1 8)` ou `(ly:make-moment 3 97)`.

Se pose alors le problème de fournir la juste durée de référence à `proportionalNotationDuration`. Il faut en l'occurrence procéder par tâtonnement, en commençant par une valeur proche de la note la plus rapide (la durée la plus courte) du morceau. Au plus la durée de référence est petite, au plus la musique sera étalée ; à l'inverse, une durée de référence élevée produira une musique resserrée.

```

\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 8)
    }
  }
}

```

```

\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 16)
    }
  }
}

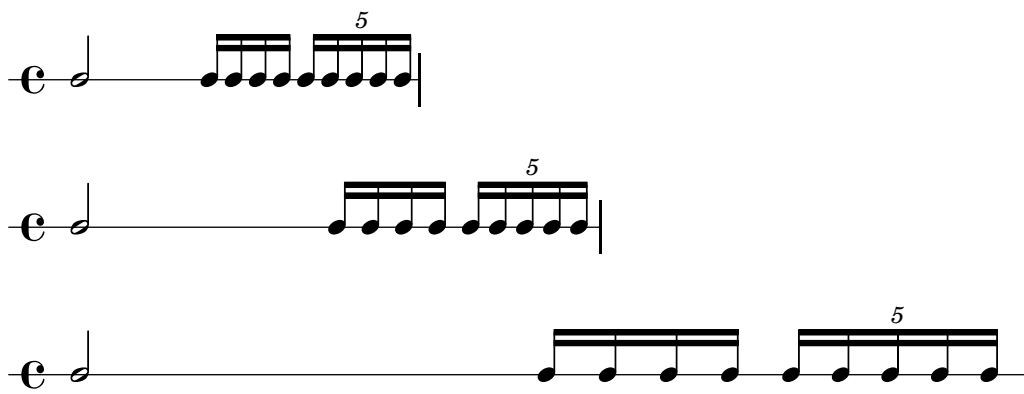
```

```

\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 32)
    }
  }
}

```

}

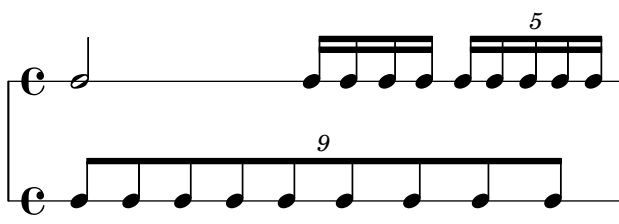


Vous ne manquerez pas de noter qu'une durée de référence trop grande, comme la croche pour la première ligne, a pour conséquence de resserrer la musique, ce qui peut aboutir à des chevauchements de têtes. Vous remarquez aussi que, par principe, la notation proportionnelle occupe beaucoup plus d'espace horizontal que l'espacement traditionnel. La notation proportionnelle met en évidence le rythme au détriment de l'espacement horizontal.

Examinons à présent le moyen d'optimiser l'espacement de nolets en tuilage.

Reprenons notre exemple de départ, avec son espacement traditionnel, et ajoutons lui une portée incluant un autre type de nolet.

```
\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
    \new RhythmicStaff {
      \times 8/9 {
        c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
      }
    }
  >>
}
```

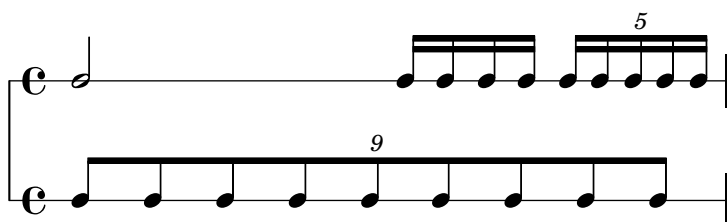


L'espacement est loin d'être idéal, pour la simple raison que l'espacement régulier des notes de la portée inférieure ne s'étire pas uniformément. Il est vrai que de telles constructions complexes en nolets sont assez rares en gravure traditionnelle, ce qui explique que les règles qu'elle applique peuvent amener à ce résultat. Le recours à `proportionalNotationDuration` permet d'arranger les choses.

```

\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
    \new RhythmicStaff {
      \times 8/9 {
        c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
      }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 20)
    }
  }
}

```



Cependant, si l'on observe de près, il est évident que les notes de la deuxième moitié du ennaolet ont tendance à s'espacer légèrement plus que celles de la première moitié. Afin d'uniformiser cet étalement, nous allons activer le **uniform-stretching**, propriété attachée au **SpacingSpanner**.

```

\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
    \new RhythmicStaff {
      \times 8/9 {
        c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
      }
    }
  >>
  \layout {
    \context {
      \Score

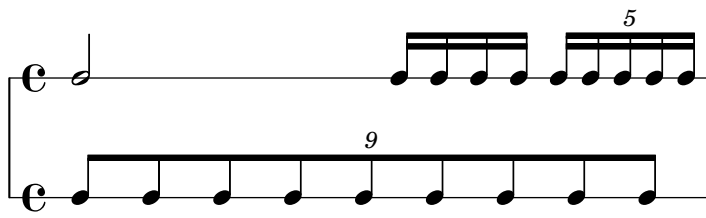
```



```

    proportionalNotationDuration = #(ly:make-moment 1 20)
    \override SpacingSpanner #'uniform-stretching = ##t
  }
}

```



L'espacement sur les deux portées est maintenant correct, les relations rythmiques sont clairement perceptibles, et nous pourrions y insérer une indication temporelle ou autre graphisme selon notre envie.

Notez bien que la prise en charge de la notation proportionnelle par LilyPond demande que, dans chaque partition, soit activée la propriété `uniform-stretching` du `SpacingSpanner`. Dans le cas contraire, utiliser `proportionalNotationDuration` aura pour conséquence, entre autres, un espacement erroné lorsque vous y aurez inséré des silences invisibles *skip*.

Le `SpacingSpanner` est en fait un objet graphique abstrait attaché au contexte `Score`. Tout comme pour la propriété `proportionalNotationDuration`, les réglages du `SpacingSpanner` peuvent se faire à trois différents endroits de votre fichier : dans un bloc `\with`, dans un bloc `\context` ou au beau milieu de la musique à l'aide de la commande `\set`.

Gardez bien à l'esprit qu'il n'y a qu'un seul `SpacingSpanner` par `Score`. Il s'ensuit que `uniform-stretching` est soit activé, soit désactivé, et dans tous les cas pour l'intégralité de la partition. Vous pourriez toutefois avoir besoin de modifier ce comportement en cours de partition, et recourir alors à l'instruction `\newSpacingSection` – pour de plus amples détails, voir la rubrique [Section 4.5.2 \[Changement d'espacement en cours de partition\]](#), page 535.

Intéressons-nous maintenant au `Separating_line_group_engraver`, qui est désactivé pour la plupart des partitions en notation proportionnelle. Voici ce qui apparaît dans une partition traditionnelle : il y a toujours un « espace préservé » juste avant la première note de chaque portée.

```

\paper {
  indent = #0
}

```

```

\new Staff {
  c'1
  \break
  c'1
}

```



Cet espace, g  r   par le `Separating_line_group_engraver`, est aussi pr  sent lorsqu'intervient un changement de m  trique, d'armure ou de clef. D  sactiver le `Separating_line_group_engraver` revient    r  duire cet espace    z  ro.

```
\paper {
  indent = #0
}

\new Staff \with {
  \remove "Separating_line_group_engraver"
} {
  c'1
  \break
  c'1
}
```



Les   l  ments non musicaux tels que m  trique, armure et clef, posent probl  me lorsqu'on travaille en notation proportionnelle. Bien qu'aucune notion de dur  e ne leur soit attach  e, ces   l  ments « consomment » de l'espace. Diff  rentes approches permettent cependant de g  rer ce probl  me.

  viter les probl  mes d'espacement avec l'armure est chose ais  e : il suffit qu'il n'y en ait pas ! C'est bien souvent le cas en musique contemporaine, o   l'on trouve le plus d'ouvrages en notation proportionnelle. Il en va de m  me pour la m  trique, et tout particuli  rement lorsque la partition comporte un quadrillage temporel ou autres graphismes. L'absence de m  trique reste cependant exceptionnelle et la plupart des partitions en notation proportionnelle laissent appara  tre quelques m  triques. Il est par contre pratiquement impossible de se passer de clef et d'alt  ration.

L'une des options permettant de s'affranchir de l'espacement d      aux   l  ments non musicaux consiste en l'activation de la propri  t   `strict-note-spacing` attach  e au `SpacingSpanner`. Observons les deux port  es suivantes :

```
\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1 16)
  c''8
  c''8
  c''8
  \clef alto
  d'8
  d'2
}

\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1 16)
  \override Score.SpacingSpanner #'strict-note-spacing = ##t
  c''8
```

```

c''8
c''8
\clef alto
d'8
d'2
}

```



Toutes deux affichent un espacement proportionnel. Cependant, la première ligne laisse apparaître un espacement plus lâche en raison de la présence d'un changement de clef. En ce qui concerne la deuxième ligne, l'espacement est strictement observé dès lors que la propriété `strict-note-spacing` a préalablement été activée. Comme vous pouvez le constater, l'activation de `strict-note-spacing` a pour conséquence que l'algorithme d'espacement ignore tout bonnement la largeur des métriques, armures, clefs et altérations.

En plus de ceux que nous venons de voir, vous trouverez d'autres réglages en usage dans la notation proportionnelle comme, entre autres,

- `\override SpacingSpanner #'strict-grace-spacing = ##t`
- `tupletFullLength = ##t`
- `\override Beam #'breakable = ##t`
- `\override Glissando #'breakable = ##t`
- `\override TextSpanner #'breakable = ##t`
- `\remove "Forbid_line_break_engraver"` dans le contexte de voix

Ces différents réglages permettent un espacement strict des notes d'ornement, d'étendre les indications de nolet afin d'indiquer de façon évidente leurs bornes et d'autoriser le tronçonnement des extenseurs à l'occasion d'un saut de ligne ou de page. Nous vous renvoyons aux différentes rubriques associées du manuel pour chacun de ces réglages.

Voir aussi

Manuel de notation : [Section 4.5.2 \[Changement d'espacement en cours de partition\]](#), page 535.

Morceaux choisis : [Section “E spacements” dans *Morceaux choisis*](#).

4.6 Réduction du nombre de pages de la partition

Vous pourriez un jour être confronté au problème suivant : l'une des pages de votre partition ne comporte que deux portées alors que – ce qui est d'autant plus frustrant – l'espace libre sur les autres pages aurait permis une distribution différente.

L'instruction `annotate-spacing` se révèle être un outil indispensable pour l'analyse des problèmes de mise en forme. Cette commande met en surimpression la valeur des différentes variables d'espacement et de mise en forme, comme nous allons le voir dans la rubrique [Section 4.6.1 \[Mise en évidence de l'espacement\]](#), page 546.

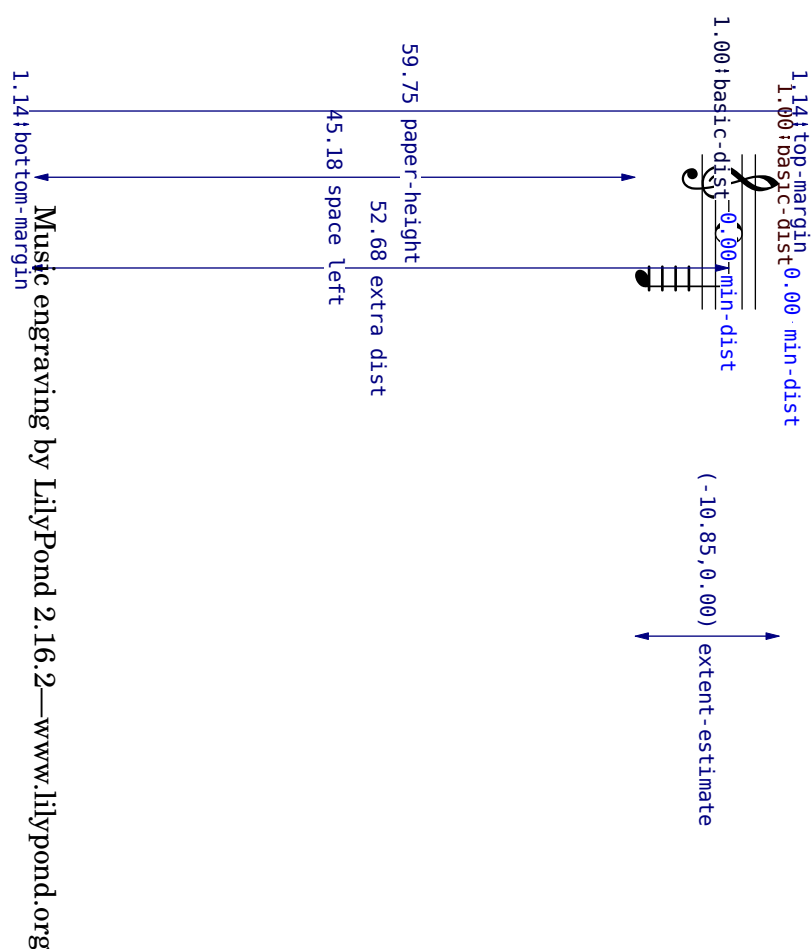
4.6.1 Mise en évidence de l'espacement

Le meilleur moyen d'appréhender les différentes variables de dimensionnement vertical sur lesquelles vous pouvez jouer au niveau de la mise en page consiste à activer, au sein du bloc `\paper`, la fonction `annotate-spacing` :

```

#(set-default-paper-size "a6" 'landscape)
\book {
  \score { { c4 } }
  \paper { annotate-spacing = ##t }
}

```



Toutes les dimensions sont exprimées en espace de portée (*staff-space*) quelle que soit l'unité mentionnée dans les blocs `\paper` ou `\layout`. Dans cet exemple, la hauteur de la feuille (`paper-height`) est de 59,75 espaces de portée (`staff-spaces`) et la taille de portée (`staff-size`) de 20 points – sa valeur par défaut. Notez que :

$$1 \text{ point} = (25,4/72,27) \text{ mm}$$

$$1 \text{ staff-space} = (\text{staff-size})/4 \text{ pts}$$

$$= \frac{(\text{staff-size})/4}{(25,4/72,27) \text{ mm}} *$$

Dans le cas qui nous occupe, un **staff-space** égale environ 1,757 millimètres. Les 59,75 **staff-spaces** de **paper-height** correspondent donc à 105 millimètres, soit la hauteur d'une feuille au format A6 à l'italienne. Les paires **(a,b)** sont des intervalles, *a* en étant l'extrémité inférieure et *b* l'extrémité supérieure.

Voir aussi

Manuel de notation : [Section 4.2.2 \[Définition de la taille de portée\]](#), page 509.

Morceaux choisis : [Section “Espaces” dans Morceaux choisis](#).

4.6.2 Modification de l'espacement

Les informations fournies par **annotate-spacing** en matière de dimensionnement vertical sont incomparables. Pour plus de détails sur les manières de modifier marges et autres variables connexes, consultez la rubrique [Section 4.1 \[Mise en forme de la page\]](#), page 496.

En dehors des marges, vous disposez de quelques moyens supplémentaires pour gagner de l'espace :

- Rapprocher les systèmes le plus possible les uns des autres, de telle sorte qu'il en tienne un maximum sur une même page, tout en les espaçant suffisamment pour éviter le blanc en bas de page.

```
\paper {
  system-system-spacing = #'((basic-distance . 0.1) (padding . 0))
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Forcer le nombre de systèmes par page. Ceci peut se révéler judicieux à deux titres. D'une part, le fait de définir un nombre de systèmes – même s'il est égal à la valeur par défaut – peut aboutir à plus de systèmes par page dans la mesure où l'une des étapes d'estimation des hauteurs est tout simplement sautée. D'autre part, réduire le nombre de systèmes par page permet d'en disposer plus sur les suivantes. Par exemple, avec un nombre par défaut de 11 systèmes par pages, l'instruction suivante le force à 10.

```
\paper {
  system-count = #10
}
```

- Forcer le nombre de pages. L'instruction suivante forcera la musique à se répartir sur deux pages.

```
\paper {
  page-count = #2
}
```

- Éviter ou réduire les objets qui augmentent la hauteur des systèmes. Par exemple, un crochet de reprise ou d'alternative consomme de l'espace. Le fait de les reporter sur plusieurs systèmes regroupés diminue d'autant l'espace disponible que si seul le premier ne comportait l'indication. Autre exemple, les indications de nuance qui se « détachent » d'un système peuvent être rapprochés de la portée :

```
e4 c g\ f c
e4 c g-\tweak #'X-offset #-2.7 -\tweak #'Y-offset #2.5 \f c
```



- Modifier l'espacement horizontal à l'aide du `SpacingSpanner`, comme indiqué à la rubrique [Section 4.5.3 \[Modification de l'espacement horizontal\]](#), page 535. Voici ce que donne l'espacement par défaut :

```
\score {
  \relative c'' {
    g4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
}
```



Par contre, le fait de modifier la valeur de la propriété `common-shortest-duration` en passant de 1/4 à 1/2 – bien que la noire soit la durée la plus courante, nous prenons une valeur plus longue – donnera un effet « resserré » à la musique :

```
\score {
  \relative c'' {
    g4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner
        #'common-shortest-duration = #(ly:make-moment 1 2)
    }
  }
}
```



La propriété `common-shortest-duration` ne peut être modifiée dynamiquement. Elle se place toujours dans un bloc `\context` et s'applique à l'intégralité de la partition.

Voir aussi

Manuel de notation : [Section 4.1 \[Mise en forme de la page\]](#), page 496, [Section 4.5.3 \[Modification de l'espacement horizontal\]](#), page 535.

Morceaux choisis : [Section “Espacements”](#) dans *Morceaux choisis*.

5 Modification des réglages prédéfinis

LilyPond est conçu pour engendrer, par défaut, des partitions de la plus haute qualité. Cependant, on peut parfois avoir à modifier cette mise en forme par défaut. Celle-ci est réglée par tout un ensemble de « leviers et manettes » plus connus sous le terme de « propriétés », dont ce chapitre ne cherche pas à faire l’inventaire exhaustif – le chapitre [Section “Retouche de partition” dans Manuel d’initiation](#) du manuel d’initiation vous en propose un aperçu. Le propos est plutôt ici de mettre en évidence les différents groupes auxquels s’apparentent ces contrôles, et d’expliquer comment trouver le bon levier pour obtenir tel ou tel effet en particulier.

Les moyens de contrôle des différents réglages sont décrits dans un document séparé, [Référence des propriétés internes](#). Ce guide répertorie toutes les variables, fonctions et autres options que LilyPond met à votre disposition. Il est consultable [en ligne](#), au format HTML ; il est également inclus dans la documentation fournie avec le logiciel.

En sous-main, LilyPond se sert du langage Scheme (un dérivé du LISP) comme infrastructure. Modifier les choix de mise en page revient à pénétrer dans les entrailles du programme, et de ce fait requiert l’emploi du Scheme. Les fragments de Scheme, dans un fichier ‘.ly’, sont introduits par le caractère *hash* (#), improprement surnommé « dièse ».¹

5.1 Contextes d’interprétation

Cette section traite des contextes.

Voir aussi

Manuel d’initiation : [Section “Contextes et graveurs” dans Manuel d’initiation](#).

Fichiers d’initialisation : ‘ly/engraver-init.ly’, ‘ly/performer-init.ly’.

Morceaux choisis : [Section “Contextes et graveurs” dans Morceaux choisis](#).

Référence des propriétés internes : [Section “Contexts” dans Référence des propriétés internes](#), [Section “Engravers and Performers” dans Référence des propriétés internes](#).

5.1.1 Tout savoir sur les contextes

Les contextes sont hiérarchisés :

Score – le père de tous les contextes

Il s’agit en l’occurrence du contexte le plus élevé, autrement dit le plus important, en matière de notation. En effet, c’est au niveau de la partition – *score* en anglais – que se gèrent le temps et la tonalité ; c’est donc là qu’il faut s’assurer que les différents éléments, tels les clefs, métriques et armures sont bien répercutés sur toutes les portées.

Dès lors que LilyPond rencontre un bloc `\score {...}` ou `\layout {...}`, se crée implicitement un contexte **Score**.

Contextes de haut niveau – les systèmes

De nombreuses partitions sont écrites sur plus d’une portée. Ces portées peuvent être regroupées de différentes manières.

StaffGroup

Le groupe de portées est attaché par un crochet, et les barres de mesure sont d’un seul tenant, de la première à la dernière portée. Le **StaffGroup** constitue le regroupement le plus simple.

ChoirStaff

¹ Le [Section “Tutoriel Scheme” dans Extension de LilyPond](#) fournit quelques notions de base pour saisir des nombres, des listes, des chaînes de caractères ou des symboles, en Scheme.

Ce regroupement est identique au **StaffGroup**, à ceci près que les barres de mesure ne traversent pas l'espace inter-portées.

GrandStaff

Le groupe de portées est attaché par une accolade sur la gauche, et les barres de mesure sont d'un seul tenant.

PianoStaff

Ce regroupement est identique au **GrandStaff**, à ceci près que le nom de l'instrument sera directement attaché au système.

Contextes de niveau intermédiaire – les portées

Staff

La portée prend en charge les clefs, barres de mesure, armures et les altérations accidentelles. Un contexte **Staff** peut contenir plusieurs contextes **Voice**.

RhythmicStaff

De même nature qu'un **Staff**, mais destiné à n'imprimer que du rythme. Quelle que soit la hauteur, les notes seront imprimées sur une même et unique ligne.

TabStaff

Ce contexte permet de générer des tablatures. La mise en forme par défaut correspond à une tablature pour guitare, sur six lignes.

DrumStaff

Contexte dévolu tout spécialement aux parties de percussion ; il peut contenir plusieurs **DrumVoice**.

VaticanaStaff

Identique au contexte **Staff**, à ceci près qu'il est tout particulièrement adapté au grégorien.

MensuralStaff

Identique au contexte **Staff**, à ceci près qu'il est tout particulièrement adapté au style mensural de musique ancienne.

Contextes de bas niveau – les voix

Les contextes de niveau « voix » initialisent un certain nombre de propriétés et activent les graveurs appropriés. S'agissant de contextes du plus bas niveau, ils ne sauraient contenir d'autre contexte.

Voice

Correspond à une voix positionnée sur une portée. Le contexte **Voice** s'occupe des indications de nuance, des hampes, des ligatures, des scripts placés au-dessus ou au-dessous de la portée, des différentes liaisons et des silences. Lorsque plusieurs voix doivent cohabiter sur la même portée, il est indispensable de les instancier explicitement.

VaticanaVoice

Fonctionnant comme le contexte **Voice**, il est tout particulièrement destiné à gérer le grégorien.

MensuralVoice

Fonctionnant comme le contexte **Voice**, il est tout particulièrement adapté aux musiques anciennes.

Lyrics

Correspond à une voix contenant des paroles. Le contexte **Lyrics** gère l'impression d'une ligne de paroles.

DrumVoice

Contexte de voix dévolu à une portée de percussions.

FiguredBass

Contexte prenant en charge les objets **BassFigure** – la basse chiffrée – créés à partir de ce qui a été saisi en mode `\figuremode`.

TabVoice

Contexte de voix dévolu au contexte **TabStaff**, il est habituellement créé implicitement.

CueVoice

Contexte de voix utilisé essentiellement dans le cadre de citations ajoutées à une portée – voir [\[Mise en forme d'une citation\]](#), page 195. Il est habituellement créé implicitement.

ChordNames

Permet d'imprimer des noms d'accord.

5.1.2 Création d'un contexte

Lorsqu'une partition ne comporte qu'une portée avec une seule voix, les contextes sont créés automatiquement. Dès que la structure s'étoffe, il devient nécessaire de les créer explicitement, en suivant l'une des trois méthodes suivantes :

- La commande la plus simple à utiliser est `\new` ; c'est aussi la plus courte. Elle se place juste avant une expression musicale et se libelle ainsi :

`\new type expression_musicale`

où *type* est le nom d'un contexte (par ex. **Staff** ou **Voice**). Cette commande crée un nouveau contexte et y interprète le contenu de l'*expression_musicale*.

C'est ce qui se passe lorsqu'une partition comporte plusieurs portées : chaque partie qui doit apparaître sur une portée spécifique est précédée d'un `\new Staff`.

```
<<
  \new Staff { c4 c }
  \new Staff { d4 d }
>>
```



La commande `\new` vous permet aussi d'attribuer un nom au contexte que vous créez.

`\new type = nom musique`

Le nom que vous spécifiez ne pourra être utilisé que s'il n'a pas déjà été attribué à un autre contexte.

- Tout comme `\new`, la commande `\context` affectera une expression musicale à un objet contextuel ; elle lui attribuera de surcroît un nom. La commande `\context` s'emploie de la façon suivante :

`\context type = nom musique`

LilyPond va dans un premier temps vérifier l'existence d'un contexte du type *type* appelé *nom*. En l'absence d'un tel contexte, LilyPond crée un nouveau contexte du nom que vous avez spécifié. Cette procédure est tout à fait pertinente lorsque vous faites appel à ce contexte particulier par la suite. Prenons le cas d'un chant : nous commençons par nommer la ligne mélodique,

```
\context Voice = "tenor" musique
```

de telle sorte que le texte s'aligne correctement sur les notes :

```
\new Lyrics \lyricsto "tenor" paroles
```

L'une des autres utilisations de contextes explicitement nommés consiste à fusionner deux expressions musicales dans un même contexte. Dans l'exemple qui suit, notes et articulations sont saisies indépendamment.

```
music = { c4 c4 }
```

```
arts = { s4-. s4-> }
```

Elles sont ensuite fusionnées par affectation au même contexte `Voice` :

```
<<
  \new Staff \context Voice = "A" \music
  \context Voice = "A" \arts
>>
```



Grâce à ce mécanisme, vous pouvez tout à fait générer une version « Urtext » (édition originale) et optionnellement ajouter distinctement des articulations à ces mêmes notes.

- Voici une troisième manière de créer un contexte :

```
\context type musique
```

Très comparable à une déclaration `\context = nom`, cette méthode permet de s'affranchir du type de contexte.

Cette variante s'utilise lorsque les expressions musicales peuvent être interprétées à différents niveaux, comme par exemple lorsque intervient la commande `\applyOutput` – pour de plus amples détails, voir [Section “Application d’une fonction à tous les objets de mise en forme” dans *Extension de LilyPond*](#). En l'absence de `\context` explicite, LilyPond considère qu'il s'agit de `Voice`.

```
\applyOutput #'contexte #fonction % s'applique à Voice
```

Vous devrez respecter ces formulations si la fonction doit s'interpréter au niveau `Score` ou `Staff` :

```
\applyOutput #'Score #fonction
```

```
\applyOutput #'Staff #fonction
```

5.1.3 Conservation d'un contexte

En règle générale, un contexte disparaît dès qu'il n'y a plus rien à faire. Autrement dit, un contexte `Voice` disparaît dès après le dernier événement qu'il contient, et un contexte `Staff` dès que les contextes `Voice` qu'il supporte ne contiennent plus rien. Ceci peut avoir des conséquences néfastes lorsqu'il est fait référence à un contexte alors disparu, comme dans le cas d'un changement de portée introduit par la commande `\change`, l'association de paroles à l'aide de la commande `\lyricsto` ou si des événements surviennent à nouveau pour ce contexte précédemment actif.

Une exception cependant à cette règle : en présence d'un contexte `Staff` ou dans une construction `<<...>>`, un seul des contextes `Voice` inclus restera actif jusqu'à la fin du contexte `Staff` ou de la construction `<<...>>`, y compris s'il y a des « trous ». Le contexte alors persistant sera le premier rencontré dans la construction `{...}` sans tenir compte des éventuels `<<...>>` qu'elle pourrait contenir.

Un contexte restera actif dès lors qu'il s'y passera toujours quelque chose. Un contexte **Staff** restera actif si l'une des voix qu'il supporte est toujours active. L'un des moyens de s'en assurer consiste à ajouter des silences invisibles parallèlement à la musique. Vous devrez les ajouter dans tous les contextes **Voice** qui doivent rester actifs. Nous vous conseillons, lorsque plusieurs voix interviennent de manière sporadique, de toutes les maintenir actives plutôt que de vous fier aux exceptions mentionnées plus haut.

Dans l'exemple suivant, les deux voix A et B sont maintenues actives jusqu'à la fin du morceau :

```
musicA = \relative c'' { d4 d d d }
musicB = \relative c'' { g4 g g g }
keepVoicesAlive = {
  <<
    \new Voice = "A" { s1*5 } % Keep Voice "A" alive for 5 bars
    \new Voice = "B" { s1*5 } % Keep Voice "B" alive for 5 bars
  >>
}

music = {
  \context Voice = "A" {
    \voiceOneStyle
    \musicA
  }
  \context Voice = "B" {
    \voiceTwoStyle
    \musicB
  }
  \context Voice = "A" { \musicA }
  \context Voice = "B" { \musicB }
  \context Voice = "A" { \musicA }
}

\score {
  \new Staff <<
    \keepVoicesAlive
    \music
  >>
}
```



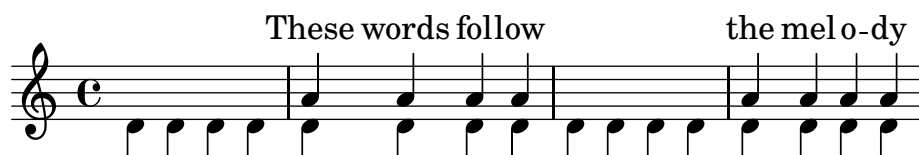
L'exemple suivant illustre la manière d'écrire selon cette méthode une mélodie discontinue à laquelle se rattachent des paroles. Dans la réalité, mélodie et accompagnement feraient l'objet de portées séparées.

```
melody = \relative c'' { a4 a a a }
accompaniment = \relative c' { d4 d d d }
words = \lyricmode { These words fol -- low the mel -- o -- dy }
\score {
  <<
    \new Staff = "music" {
```

```

<<
  \new Voice = "melody" {
    \voiceOne
    s1*4 % Keep Voice "melody" alive for 4 bars
  }
  {
    \new Voice = "accompaniment" {
      \voiceTwo
      \accompaniment
    }
    <<
      \context Voice = "melody" { \melody }
      \context Voice = "accompaniment" { \accompaniment }
    >>
    \context Voice = "accompaniment" { \accompaniment }
    <<
      \context Voice = "melody" { \melody }
      \context Voice = "accompaniment" { \accompaniment }
    >>
  }
>>
}
\new Lyrics \with { alignAboveContext = #"music" }
\lyricsto "melody" { \words }
>>
}

```



Une autre méthode, qui s'avère plus productive dans nombre de cas, consiste à maintenir active la ligne mélodique en y insérant des silences invisibles tout au long de l'accompagnement :

```

melody = \relative c'' {
  s1 % skip a bar
  a4 a a a
  s1 % skip a bar
  a4 a a a
}
accompaniment = \relative c' {
  d4 d d d
  d4 d d d
  d4 d d d
  d4 d d d
}
words = \lyricmode { These words fol -- low the mel -- o -- dy }

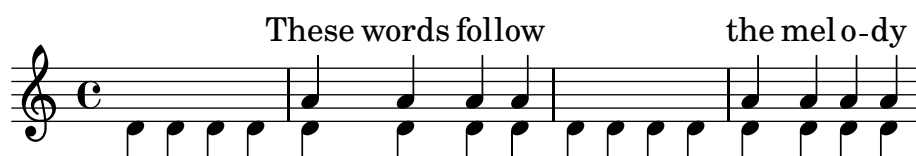
\score {
  <<
    \new Staff = "music" {
      <<

```

```

\new Voice = "melody" {
  \voiceOne
  \melody
}
\new Voice = "accompaniment" {
  \voiceTwo
  \accompaniment
}
>>
}
\new Lyrics \with { alignAboveContext = #"music" }
\lyricsto "melody" { \words }
>>
}

```



5.1.4 Modification des greffons de contexte

Les contextes, tels que **Score** ou **Staff**, ne contiennent pas que des propriétés ; ils mettent également en œuvre certains sous-programmes (*plug-ins* pour employer le terme consacré) nommés « graveurs » (*engravers* pour reprendre le terme anglais). Ces sous-programmes sont chargés de créer les différents éléments de notation : on trouve ainsi dans le contexte **Voice** un graveur **Note_heads_engraver**, chargé des têtes de notes et, dans le contexte **Staff**, un graveur **Key_engraver**, chargé de l'armure.

Vous trouverez une description exhaustive de chaque graveur dans Référence des propriétés internes \mapsto Translation \mapsto Engravers. Chaque contexte mentionné dans Référence des propriétés internes \mapsto Translation \mapsto Context. répertorie les graveurs mis en œuvre.

On peut faire, au moyen de ces graveurs, sa propre « cuisine », en modifiant les contextes à volonté.

Lorsqu'un contexte est créé, par la commande **\new** ou **\context**, on peut y adjoindre un bloc **\with** (en anglais « avec »), dans lequel il est possible d'ajouter (commande **\consists**) ou d'enlever (commande **\remove**) des graveurs :

```

\new contexte \with {
  \consists ...
  \consists ...
  \remove ...
  \remove ...
  etc.
}
{
  ..musique..
}

```

Ici les points de suspension ... devront être remplacés par le nom des graveurs désirés. Dans l'exemple suivant, on enlève du contexte **Staff**, la métrique (graveur **Time_signature_engraver**) et la clef (graveur **Clef_engraver**).

```

<<
\new Staff {

```

```

    f2 g
  }
  \new Staff \with {
    \remove "Time_signature_engraver"
    \remove "Clef_engraver"
  } {
    f2 g2
  }
>>

```



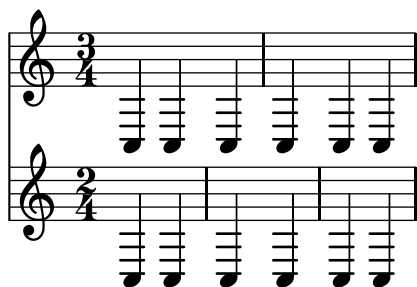
La clef et le chiffre de mesure ont disparu de la deuxième portée. C'est une méthode quelque peu radicale, puisqu'elle affectera toute la portée jusqu'à la fin de la partition. L'espacement s'en trouve également affecté, ce qui peut être ou non l'effet recherché. Une méthode plus sophistiquée aurait été de rendre ces objets transparents (voir [Section "Visibilité et couleur des objets" dans Manuel d'initiation](#)).

Dans l'exemple suivant, voici une mise en pratique plus utile. En temps normal, les barres de mesure et la métrique sont synchronisées verticalement dans toute la partition. Les graveurs qui en sont responsables se nomment `Timing_translator` et `Default_bar_line_engraver`. En les enlevant du contexte `Score` pour les attribuer au contexte `Staff`, chaque portée peut désormais avoir sa propre métrique.

```

\score {
  <<
    \new Staff \with {
      \consists "Timing_translator"
      \consists "Default_bar_line_engraver"
    } {
      \time 3/4
      c4 c c c c c
    }
    \new Staff \with {
      \consists "Timing_translator"
      \consists "Default_bar_line_engraver"
    } {
      \time 2/4
      c4 c c c c c
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Timing_translator"
      \remove "Default_bar_line_engraver"
    }
  }
}

```



Problèmes connus et avertissements

L'ordre dans lequel les graveurs sont spécifiés correspond à leur ordre d'apparition dans le processus d'élaboration de la partition. En règle générale, l'ordre dans lequel les graveurs sont mentionnés importe peu. Il se peut toutefois qu'un graveur écrive une propriété qui sera interprétée par un autre, ou qu'un graveur crée un objet graphique qui sera traité par un autre ; l'ordre d'apparition de ces graveurs prendra alors tout son importance.

Pour information, les ordonnancements suivants sont importants :

- le `Bar_engraver` devrait toujours être le premier ;
- le `New_fingering_engraver` doit toujours précéder le `Script_column_engraver` ;
- le `Timing_translator` doit toujours précéder le `Bar_number_engraver`.

Voir aussi

Fichiers d'initialisation : `'ly/engraver-init.ly'`.

5.1.5 Modification des réglages par défaut d'un contexte

Les propriétés des contextes et objets graphiques se modifient à l'aide des commandes `\set` et `\override`, comme indiqué à la rubrique [Section 5.3 \[Modification de propriétés\]](#), page 569. Ces commandes créent des événements musicaux qui feront que la modification produira ses effets dès l'instant où la musique est traitée.

Le propos est ici de voir comment modifier les valeurs *par défaut* des propriétés de contexte ou d'objet graphique dès la création de ces contextes. Deux manières de procéder sont envisageables : l'une consiste à modifier les valeurs pour tous les contextes d'un même type, l'autre s'attache à adapter les valeurs par défaut d'une instance particulière d'un contexte.

Modification de tous les contextes d'un même type

La personnalisation des réglages par défaut d'un contexte, qu'il s'agisse de `Score`, `Staff` ou `Voice`, peut se réaliser indépendamment de la musique dans un bloc `\layout` – placé dans le bloc `\score` auquel ces modifications doivent s'appliquer – au moyen d'un bloc `\context`.

```
\layout {
  \context {
    \Voice
    [réglage de contexte pour tous les contextes Voice]
  }
  \context {
    \Staff
    [réglage de contexte pour tous les contextes Staff]
  }
}
```

La spécification des adaptations peut se faire de différentes manières :

- à l'aide d'une commande `\override`, sans lui adjoindre le nom du contexte :

```
\score {
  \relative c'' {
```

```

a4~"Hampes épaisses" a a a
a4 a a\ff a
}
\layout {
  \context {
    \Staff
    \override Stem #'thickness = #4.0
  }
}

```

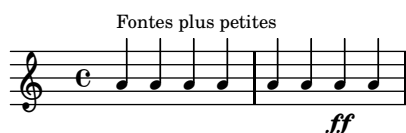


- en définissant directement une propriété de contexte :

```

\score {
  \relative c'' {
    a4~"Fontes plus petites" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Staff
      fontSize = #-4
    }
  }
}

```



- à l'aide d'une commande prédéfinie comme `\dynamicUp`, ou bien une expression musicale telle que `\accidentalStyle "dodecaphonic"` :

```

\score {
  \relative c'' {
    a4~"Nuance en surplomb" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Voice
      \dynamicUp
    }
    \context {
      \Staff
      \accidentalStyle "dodecaphonic"
    }
  }
}

```


}

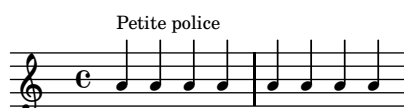
Nuance en surplomb



- à l'aide d'une variable personnalisée contenant un bloc `\with` : pour de plus amples informations sur le bloc `\with`, voir [\[Modification d'un contexte particulier\]](#), page 560.

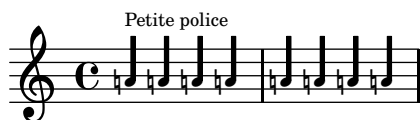
```
StaffDefaults = \with {
  fontSize = #-4
}
```

```
\score {
  \new Staff {
    \relative c'' {
      a4^"Petite police" a a a
      a4 a a a
    }
  }
  \layout {
    \context {
      \Staff
      \StaffDefaults
    }
  }
}
```



Les instructions destinées à modifier les propriétés peuvent se placer dans un bloc `\layout` sans pour autant être incluses dans un bloc `\context`. Expliciter des réglages de la sorte équivaut à inclure les commandes de modification des propriétés au début de chacun des contextes du type en question. Lorsque le contexte n'est pas spécifié, *tous* les contextes de bas niveau seront affectés – voir [\[Contextes de bas niveau – les voix\]](#), page 550. La syntaxe appropriée répond aux mêmes critères que si la commande était écrite dans le flot musical.

```
\score {
  \new Staff {
    \relative c'' {
      a4^"Petite police" a a a
      a4 a a a
    }
  }
  \layout {
    \accidentalStyle "dodecaphonic"
    \set fontSize = #-4
    \override Voice.Stem #'thickness = #4.0
  }
}
```



Modification d'un contexte particulier

Dans le cas d'un contexte pris individuellement, ses propriétés se modifient à l'aide d'un bloc `\with`. Toutes les autres instances de contexte appartenant au même type seront affectés des réglages prédéfinis par LilyPond, modifiés le cas échéant par un bloc `\layout`. Le bloc `\with` se place directement à la suite de la commande `\new type-de-contexte`.

```
\new Staff
\with {
  [réglages pour ce contexte pris individuellement]
} {
  ...
}
```

La spécification des adaptations peut se faire de différentes manières :

- à l'aide d'une commande `\override`, sans lui adjoindre le nom du contexte :

```
\score {
  \new Staff {
    \new Voice
    \with {
      \override Stem #'thickness = #4.0
    }
    {
      \relative c'' {
        a4^"Hampes épaisses" a a a
        a4 a a a
      }
    }
  }
}
```



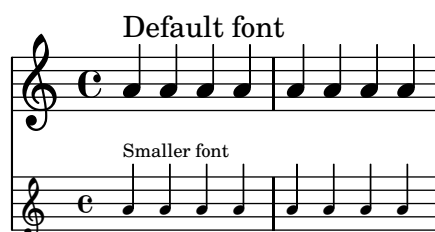
- en définissant directement une propriété de contexte :

```
\score {
  <<
    \new Staff {
      \relative c'' {
        a4^"Default font" a a a
        a4 a a a
      }
    }
    \new Staff
    \with {
      fontSize = #-4
    } {
      \relative c'' {
        a4^"Smaller font" a a a
        a4 a a a
      }
    }
  }
}
```

```

    }
  }
  >>
}

```



- à l'aide d'une commande prédéfinie comme `\dynamicUp` :

```

\score {
  <<
    \new Staff {
      \new Voice {
        \relative c'' {
          a4^"Nuances en dessous" a a a
          a4 a a\ff a
        }
      }
    }
    \new Staff
    \with { \accidentalStyle "dodecaphonic" }
    {
      \new Voice
      \with { \dynamicUp }
      {
        \relative c'' {
          a4^"Nuances en surplomb" a a a
          a4 a a\ff a
        }
      }
    }
  }
  >>
}

```



Ordre de préséance

La valeur d'une propriété qui doit s'appliquer à un instant particulier est déterminée comme suit :

- s'il y a une instruction `\override` ou `\set` active dans le flot d'information, sa valeur s'applique,
- en l'absence de quoi sera utilisée la valeur par défaut telle que définie dans une clause `\with` stipulée à l'initialisation du contexte,
- en l'absence de quoi sera retenue la valeur par défaut issue du bloc `\context` approprié le plus récent dans le bloc `\layout`,
- en l'absence de quoi s'appliqueront les réglages prédéfinis de LilyPond.

Voir aussi

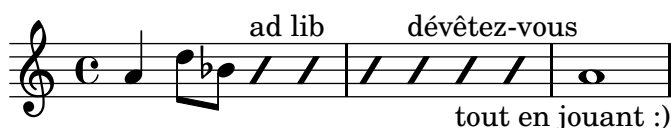
Manuel d'initiation : [Section "Modification des propriétés d'un contexte"](#) dans *Manuel d'initiation*.

Manuel de notation : [Section 5.1.1 \[Tout savoir sur les contextes\]](#), page 549, [\[Contextes de bas niveau – les voix\]](#), page 550, [Section 5.3.2 \[La commande de fixation \(set\)\]](#), page 569, [Section 5.3.3 \[La commande de dérogation \(override\)\]](#), page 571, [Section 4.2.1 \[Le bloc \layout\]](#), page 507.

5.1.6 Définition de nouveaux contextes

Les contextes tels que **Staff** ou **Voice** sont faits de briques de construction empilées. En combinant divers graveurs, il est possible de créer de nouveaux types de contextes.

Dans l'exemple suivant on construit, de zéro, un nouveau contexte très semblable à **Voice**, mais qui n'imprime que des têtes de notes en forme de barre oblique au centre de la portée. Un tel contexte peut servir, par exemple, à indiquer un passage improvisé dans un morceau de jazz.



On a rassemblé les réglages dans un bloc `\context`, lui-même placé dans le bloc `\layout` :

```
\layout {
  \context {
    ...
  }
}
```

En lieu et place des points (...), voici les éléments à saisir :

Tout d'abord, il convient de donner un nom à notre nouveau contexte :

```
\name ImproVoice
```

Comme il est très semblable à **Voice**, nous souhaitons que toutes les commandes associées au **Voice** déjà existant restent valables. D'où nécessité de la commande `\alias`, qui va l'associer au contexte **Voice** :

```
\alias Voice
```

Ce contexte doit pouvoir imprimer des notes et des indications textuelles ; on ajoute donc les graveurs appropriés :

```
\consists "Note_heads_engraver"
\consists "Text_engraver"
```

Cependant, on veut que les notes s'affichent toutes au centre de la portée :

```
\consists "Pitch_squash_engraver"
squashedPosition = #0
```

Le graveur [Section "Pitch_squash_engraver"](#) dans *Référence des propriétés internes* intercepte les notes créées par le [Section "Note_heads_engraver"](#) dans *Référence des propriétés*

internes, et les « écrase » pour qu'elles aient toutes la même position verticale, définie par `squashedPosition` : ici il s'agit de la valeur 0, c'est-à-dire la ligne du milieu.

On veut que les notes aient la forme d'une barre oblique, sans aucune hampe :

```
\override NoteHead #'style = #'slash
\override Stem #'transparent = ##t
\override Flag #'transparent = ##t
```

Afin que tous ces graveurs puissent travailler de concert, on leur adjoint un sous-programme spécial, introduit par la commande `\type` : il s'agit de `Engraver_group`,

```
\type "Engraver_group"
```

Récapitulons ; on se retrouve avec le bloc suivant :

```
\context {
  \name ImproVoice
  \type "Engraver_group"
  \consists "Note_heads_engraver"
  \consists "Text_engraver"
  \consists "Pitch_squash_engraver"
  squashedPosition = #0
  \override NoteHead #'style = #'slash
  \override Stem #'transparent = ##t
  \override Flag #'transparent = ##t
  \alias Voice
}
```

Ce n'est pas tout. En effet, on veut intégrer le nouveau contexte `ImproVoice` dans la hiérarchie des contextes. Tout comme le contexte `Voice`, sa place est au sein du contexte `Staff`. Nous allons donc modifier le contexte `Staff`, au moyen de la commande `\accepts` :

```
\context {
  \Staff
  \accepts ImproVoice
}
```

Le contraire de `\accepts` est `\denies` ; il est parfois utile lorsque l'on recycle des définitions de contextes déjà existantes.

Enfin, tout cela doit prendre place dans le bloc `\layout`, comme ici :

```
\layout {
  \context {
    \name ImproVoice
    ...
  }
  \context {
    \Staff
    \accepts "ImproVoice"
  }
}
```

On peut alors saisir la musique, comme dans l'exemple plus haut :

```
\relative c'' {
  a4 d8 bes8
  \new ImproVoice {
    c4~"ad lib" c
    c4 c~"dévêtez-vous"
    c c_"tout en jouant :)"
  }
}
```

```

}
a1
}

```

5.1.7 Ordonnancement des contextes

Les contextes viennent en principe se positionner selon leur ordre d'apparition dans le fichier source. Lorsque plusieurs contextes sont imbriqués, le contexte englobant supportera les différents contextes mentionnés dans le fichier source, à la stricte condition qu'ils soient dûment « agréés ». Les contextes imbriqués qui ne font pas partie de la « liste d'agréments » du contexte englobant se retrouveront en dessous de celui-ci au lieu d'y être imbriqués.

La liste des « agréments » d'un contexte se gère à l'aide des instructions `\accepts` et `\denies` – `\accepts` pour ajouter un contexte à la liste, et `\denies` pour retirer l'agrément. Il est par exemple peu conventionnel que les accords nommés apparaissent dans un contexte `Staff` ; autrement dit, le contexte `ChordNames` ne fait pas partie de la « liste d'agréments » du contexte `Staff` par défaut. Néanmoins, et s'il devait en être ainsi, vous pourriez le spécifier.

```

\score {
  \new Staff {
    c' d' e' f'
    \chords { d1:m7 b1:min7.5- }
  }
}

```



```

\score {
  \new Staff {
    c' d' e' f'
    \chords { d1:m7 b1:min7.5- }
  }
  \layout {
    \context {
      \Staff
      \accepts "ChordNames"
    }
  }
}

```



L'instruction `\denies` permet, lorsqu'un nouveau contexte reprend les définitions d'un contexte existant, d'en ajuster les composantes. C'est par exemple le cas du contexte `VaticanaStaff`, réplique du contexte `Staff` au sein duquel le contexte `VaticanaVoice` se substitue au contexte `Voice` dans la « liste d'agrément ».

Gardez à l'esprit que, face à une instruction qui ne s'appliquerait à aucun contexte déjà existant, LilyPond créera un nouveau contexte implicite. Ceci peut engendrer une nouvelle portée ou une autre partition.

Il arrive qu'un contexte ne doive exister que pendant un court instant, ce qui est le cas par exemple pour une *ossia*. Le plus simple consiste alors à initialiser la définition d'un contexte à l'endroit approprié, en parallèle avec le fragment correspondant dans la musique principale. Ce contexte temporaire sera par défaut positionné sous les autres contextes existants. Le repositionner au-dessus du contexte « principal » demande de le définir ainsi :

```
\new Staff \with { alignAboveContext = #"principal" }
```

Il en va de même pour les contextes temporaires de paroles au sein d'un système à plusieurs portées comme un **ChoirStaff** lorsque, par exemple, un couplet supplémentaire apparaît à l'occasion d'une reprise. Ce contexte de paroles temporaire se place par défaut sous les portées inférieures. Lui adjoindre une instruction **alignBelowContext** dès son initialisation permet de l'accoler au contexte de paroles (nommé) qui contient le premier couplet.

Des exemples de repositionnement de contexte temporaire sont disponibles aux rubriques [Section “Expressions musicales imbriquées”](#) dans *Manuel d'initiation*, [Section 1.6.2 \[Modification de portées individuelles\]](#), page 178 et [Section 2.1.2 \[Situations particulières en matière de paroles\]](#), page 250.

Voir aussi

Manuel d'initiation : [Section “Expressions musicales imbriquées”](#) dans *Manuel d'initiation*.

Manuel de notation : [Section 1.6.2 \[Modification de portées individuelles\]](#), page 178, [Section 2.1.2 \[Situations particulières en matière de paroles\]](#), page 250.

Manuel d'utilisation : [Section “Apparition d'une portée supplémentaire”](#) dans *Utilisation des programmes*.

Fichiers d'initialisation : 'ly/engraver-init.ly'.

5.2 En quoi consiste la référence des propriétés internes

5.2.1 Navigation dans les références du programme

Comment, par exemple, déplacer le doigté dans le fragment suivant ?

```
c-2
\stemUp
f
```



Sur la page de la documentation relative aux doigtés, c'est-à-dire [\[Doigtés\]](#), page 201, se trouve l'indication suivante :

Voir aussi

Référence des propriétés internes : [Section “Fingering”](#) dans *Référence des propriétés internes*.

Ladite référence est disponible au format HTML, ce qui rend la navigation bien plus aisée. Il est possible soit de la lire en ligne, soit de la télécharger dans ce format. La démarche présentée ici sera plus difficile à comprendre dans un document au format PDF.

Suivons le lien [Section “Fingering”](#) dans *Référence des propriétés internes*. En haut de la nouvelle page, on peut lire

Fingering objects are created by: [Section “Fingering-engraver”](#) dans *Référence des propriétés internes* and [Section “New_fingering-engraver”](#) dans *Référence des propriétés internes*.

En d'autres termes, *Les indications de doigtés (Fingering en anglais) sont créées par les graveurs Section "Fingering-engraver" dans Référence des propriétés internes et Section "New-fingering-engraver" dans Référence des propriétés internes.*

En suivant derechef les liens propres à la référence du programme, on suit en fait le cheminement qui aboutit à la création de la partition :

- Section "Fingering" dans *Référence des propriétés internes*: Section "Fingering" dans *Référence des propriétés internes* objects are created by: Section "Fingering-engraver" dans *Référence des propriétés internes*
- Section "Fingering-engraver" dans *Référence des propriétés internes*: Music types accepted: Section "fingering-event" dans *Référence des propriétés internes*
- Section "fingering-event" dans *Référence des propriétés internes*: Music event type **fingering-event** is in Music expressions named Section "FingeringEvent" dans *Référence des propriétés internes*

Ce cheminement se produit, bien sûr, en sens inverse : nous sommes ici partis du résultat, et avons abouti aux événements (en anglais *Events*) engendrés par le fichier d'entrée. L'inverse est également possible : on peut partir d'un événement, et suivre le cheminement de LilyPond qui aboutit à la création d'un ou plusieurs objets graphiques.

La référence des propriétés internes peut également se parcourir comme un document normal. On y trouve des chapitres tels que **Music definitions** Section "Translation" dans *Référence des propriétés internes*, ou encore Section "Backend" dans *Référence des propriétés internes*. Chaque chapitre recense toutes les définitions employées, et les propriétés sujettes à ajustements.

La Référence des propriétés internes n'est pas traduite en français – notamment du fait qu'elle est en évolution constante, tout comme LilyPond. En revanche, les termes musicaux font l'objet d'un **Section "glossaire" dans Glossaire** fort utile pour les utilisateurs francophones.

5.2.2 Interfaces de rendu

Tous les éléments de notation sont considérés comme des objets graphiques (en anglais *Graphical Object*, d'où le diminutif *Grob*). Chaque objet est doté d'un certain nombre de propriétés (l'épaisseur du trait, l'orientation, etc.), et lié à d'autres objets. Le fonctionnement de ces objets est décrit en détail dans **Section "grob-interface" dans Référence des propriétés internes**.

Prenons l'exemple des doigtés (en anglais *Fingering*). La page **Fingering** de la Référence des propriétés internes établit une liste de définitions propres à ce type d'objet :

padding (dimension, in staff space):

0.5

Ce qui signifie que les doigtés doivent être maintenus à une certaine distance (*padding*) des notes : 0,5 unités *staff-space* (espace de portée).

Chaque objet peut avoir plusieurs attributs, en tant qu'élément typographique ou musical. Ainsi, un doigté (objet *Fingering*) possède les attributs suivants :

- Sa taille ne dépend pas de l'espacement horizontal, contrairement aux liaisons ou ligatures.
- C'est du texte – un texte vraiment court, certes.
- Ce texte est imprimé au moyen d'une fonte, contrairement aux liaisons ou ligatures.
- Sur l'axe horizontal, le centre de ce symbole doit être aligné avec le centre de la note.
- Sur l'axe vertical, le symbole doit être proche de la note et de la portée.
- Sur l'axe vertical encore, il doit également s'ordonner avec les éventuels autres symboles, ponctuations ou éléments textuels.

Faire appliquer ces différents attributs est le rôle des *interfaces*, que l'on trouve en bas de la page **Section "Fingering" dans Référence des propriétés internes**.

This object supports the following interfaces: Section “item-interface” dans *Référence des propriétés internes*, Section “self-alignment-interface” dans *Référence des propriétés internes*, Section “side-position-interface” dans *Référence des propriétés internes*, Section “text-interface” dans *Référence des propriétés internes*, Section “text-script-interface” dans *Référence des propriétés internes*, Section “font-interface” dans *Référence des propriétés internes*, Section “finger-interface” dans *Référence des propriétés internes*, and Section “grob-interface” dans *Référence des propriétés internes*.

En français,

Cet objet admet les interfaces suivantes :

Suit la liste des interfaces en question, présentées comme autant de liens qui conduisent aux pages dédiées à chacune d’entre elles. Chaque interface est dotée d’un certain nombre de propriétés, dont certaines peuvent être modifiées, et d’autres non (les *Internal properties*, ou propriétés internes).

Pour aller encore plus loin, plutôt que de simplement parler de l’objet **Fingering**, ce qui ne nous avance pas à grand chose, on peut aller explorer son âme même, dans les fichiers source de LilyPond (voir Section “Autres sources de documentation” dans *Manuel d’initiation*), en l’occurrence le fichier ‘scm/define-grobs.scm’ :

```
(Fingering
 . ((padding . 0.5)
    (avoid-slur . around)
    (slur-padding . 0.2)
    (staff-padding . 0.5)
    (self-alignment-X . 0)
    (self-alignment-Y . 0)
    (script-priority . 100)
    (stencil . ,ly:text-interface::print)
    (direction . ,ly:script-interface::calc-direction)
    (font-encoding . fetaText)
    (font-size . -5) ; don't overlap when next to heads.
    (meta . ((class . Item)
             (interfaces . (finger-interface
                           font-interface
                           text-script-interface
                           text-interface
                           side-position-interface
                           self-alignment-interface
                           item-interface))))))
```

...où l’on découvre que l’objet **Fingering** n’est rien de plus qu’un amas de variables et de réglages. La page de la Référence des propriétés internes est en fait directement engendrée par cette définition.

5.2.3 Détermination de la propriété d’un objet graphique (grob)

Nous voulions changer la position du chiffre **2** dans le fragment suivant :

```
c-2
\stemUp
f
```



Dans la mesure où le **2** est placé, verticalement, à proximité de la note qui lui correspond, nous allons devoir trouver l'interface en charge de ce placement, qui se trouve être **side-position-interface**. Sur la page de cette interface, on peut lire :

side-position-interface

Position a victim object (this one) next to other objects (the support). The property **direction** signifies where to put the victim object relative to the support (left or right, up or down?)

Ce qui signifie

side-position-interface

Placer l'objet affecté à proximité d'autres objets. La propriété **direction** indique où positionner l'objet (à droite ou à gauche, en haut ou en bas).

En dessous de cette description se trouve décrite la variable **padding** :

padding (dimension, in staff space)

Add this much extra space between objects that are next to each other.

Ce qui signifie

Ajouter tel espace supplémentaire entre des objets proches les uns des autres.

En augmentant la valeur de **padding**, on peut donc éloigner le doigté de la note. La commande suivante insère trois unités d'espace vide entre la note et le doigté :

```
\once \override Voice.Fingering #'padding = #3
```

En ajoutant cette commande avant la création du doigté (de l'objet **Fingering**), donc avant **c2**, on obtient le résultat suivant :

```
\once \override Voice.Fingering #'padding = #3
```

```
c-2
```

```
\stemUp
```

```
f
```



Dans le cas présent, le réglage intervient dans le contexte **Voice**, ce qui pouvait également se déduire de la Référence des propriétés internes, où la page du graveur **Section “Fingering-engraver”** dans *Référence des propriétés internes* indique :

Fingering-engraver is part of contexts: ... **Section “Voice”** dans *Référence des propriétés internes*

Ce qui signifie

Le graveur Fingering-engraver fait partie des contextes : ... **Section “Voice”** dans *Référence des propriétés internes*

5.2.4 Conventions de nommage

Afin de s'y retrouver plus aisément et d'éviter les erreurs de frappe, voici quelques conventions en matière de nommage :

- fonctions scheme : minuscule-avec-trait-d-union (y compris noms en mot-unique)
- fonctions scheme : ly:plus-style-scheme
- événements, classes et propriétés musicaux : identique-aux-fonctions-scheme
- interfaces d'objet graphique : style-scheme
- propriétés d'arrière plan : style-scheme (mais X et Y pour les axes)

- contextes (ainsi que MusicExpressions et grobs) : Capitale initiale ou Camélisation (Camel-Case)
- propriétés de contexte : minusculeSuivieDeCamélisation
- graveurs : Capitale_initiale_puis_minuscules_séparées_par_un_souligné

Les questions que vous devez vous poser sont :

- Qu'est-ce qui relève des conventions, et qu'est-ce qui relève de la règle ?
- Qu'est-ce qui relève des règles du langage sous-jacent, et qu'est-ce qui est propre à LilyPond ?

5.3 Modification de propriétés

5.3.1 Vue d'ensemble de la modification des propriétés

Chaque contexte est chargé de créer plusieurs types d'objets graphiques. Il contient également les réglages nécessaires pour chacun de ces objets. Si l'on modifie ces réglages, les objets n'auront plus la même apparence.

Les contextes comportent deux types différents de propriétés : des propriétés de contexte et des propriétés d'objet graphique. Les propriétés de contexte sont celles qui s'appliqueront globalement au contexte en tant que tel ; elles gèrent la manière dont le contexte apparaîtra. Les propriétés d'objet graphique, par contre, s'appliquent à des types particuliers d'objet qui apparaissent dans le contexte en question.

Les commandes `\set` et `\unset` permettent de modifier les valeurs des propriétés de contexte. Les commandes `\override` et `\revert` permettent de modifier les valeurs des propriétés des objets graphiques.

Voir aussi

Référence des propriétés internes : *Section “Backend” dans Référence des propriétés internes*, *Section “All layout objects” dans Référence des propriétés internes*, *Section “OverrideProperty” dans Référence des propriétés internes*, *Section “RevertProperty” dans Référence des propriétés internes*, *Section “PropertySet” dans Référence des propriétés internes*.

Problèmes connus et avertissements

La sous-couche Scheme ne vérifie pas la saisie des propriétés de façon très stricte. Des références cycliques dans des valeurs Scheme peuvent de ce fait interrompre ou faire planter le programme – ou bien les deux.

5.3.2 La commande de fixation `\set`

Chaque contexte peut avoir plusieurs **propriétés**, c'est-à-dire des variables qu'il inclut. Ces dernières peuvent être modifiées « à la volée », c'est-à-dire pendant que la compilation s'accomplit. C'est ici le rôle de la commande `\set`.

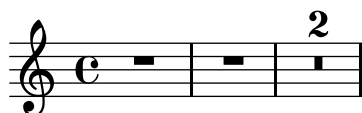
`\set contexte.propriété = #valeur`

Dans la mesure où *valeur* est constituée d'un objet Scheme, elle doit être précédée du caractère `#`.

Les propriétés des contextes se libellent sous la forme `minusculeMajuscule`. Leur rôle consiste principalement à traduire la musique en notation : par exemple, `localKeySignature` déterminera quand imprimer une altération accidentelle, et `measurePosition` quand imprimer une barre de mesure. La valeur des propriétés des contextes peuvent évoluer au fur et à mesure que l'on avance dans le morceau – `measurePosition` en est l'illustration parfaite.

Ainsi la propriété de contexte `skipBars` permet de condenser les mesures vides de notes, en des silences multimesures. Il s'agit d'un objet Scheme, auquel on attribue la valeur booléenne « vrai », c'est-à-dire la lettre `#t` pour « True » en anglais :

```
R1*2
\set Score.skipBars = ##t
R1*2
```



Si l'argument *contexte* n'est pas spécifié, alors la propriété cherchera à s'appliquer dans le contexte le plus restreint où elle est employée : le plus souvent **ChordNames**, **Voice** ou **Lyrics**.

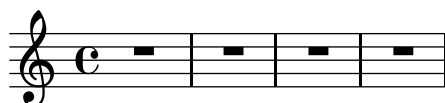
```
\set Score.autoBeaming = ##f
<<
{
  e8 e e e
  \set autoBeaming = ##t
  e8 e e e
} \ {
  c8 c c c c8 c c c
}
>>
```



Ce changement étant appliqué « à la volée », il n'affecte que le second groupe de notes.

Notez que le contexte le plus restreint n'est pas toujours le bon, et peut ne pas contenir la propriété qui vous intéresse : ainsi, la propriété **skipBars**, évoquée plus haut, ne relève pas du contexte **Voice**, mais du contexte **Score** – le code suivant ne fonctionnera pas.

```
R1*2
\set skipBars = ##t
R1*2
```



Les contextes s'organisent de façon hiérarchique : aussi, lorsqu'un contexte de niveau supérieur est spécifié (par exemple **Staff**), la propriété sera modifiée dans tous les contextes inférieurs (tous les contextes **Voice**, par exemple) qu'il contient.

La commande **\unset** permet d'annuler la définition d'une propriété :

```
\unset contexte.propriété
```

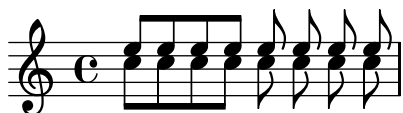
si et seulement si cette *propriété* a été définie dans ce *contexte* précis. En d'autres termes, la commande **\unset** doit impérativement affecter le même contexte que la commande **\set** d'origine, même en cas d'imbrication.

```
\set Score.autoBeaming = ##t
<<
{
  \unset autoBeaming
  e8 e e e
  \unset Score.autoBeaming
```

```

      e8 e e e
    } \ {
      c8 c c c c8 c c c
    }
  >>

```



Si l'on se trouve dans le contexte le plus restreint, il n'est pas obligatoire, là encore, de spécifier le *contexte*. Ainsi, les deux lignes suivantes

```

\set Voice.autoBeaming = ##t
\set autoBeaming = ##t

```

sont équivalentes elles apparaissent dans un contexte *Voice*.

Pour modifier une propriété de façon à ce que l'accommodement ne s'applique qu'une seule fois, il convient d'ajouter l'instruction `\once` à la commande `\set` :

```

c4
\once \set fontSize = #4.7
c4
c4

```



Ici le changement de taille est annulé aussitôt après la note concernée.

La référence des propriétés internes contient une description exhaustive de toutes les propriétés, contexte par contexte : voir Translation \mapsto Tunable context properties.

Voir aussi

Référence des propriétés internes : [Section “Tunable context properties”](#) dans *Référence des propriétés internes*.

5.3.3 La commande de dérogation `\override`

La commande `\override` permet de modifier la mise en forme des objets graphiques. Les descriptions d'objet graphique, dont les noms commencent par une majuscule, puis comprennent une ou plusieurs majuscules (de style *TotoTata*), contiennent les réglages « par défaut » pour les objets graphiques. Ces réglages sont sous forme de liste Scheme ; on peut les consulter dans le fichier `'scm/define-grobs.scm'`.

`\override` est en fait un raccourci :

```

\override contexte.NomObjet #'propriété = #valeur

```

plus ou moins équivalent à

```

\set contexte.NomObjet =
  #(cons (cons 'propriété valeur)
    <valeur antérieure de contexte.NomObjet>)

```

La valeur de `contexte.NomObjet` (une liste associative, ou *alist*) permet d'initialiser les propriétés des objets un par un. Les objets eux-mêmes ont leurs propriétés, dont les noms, dans

la tradition du langage Scheme, comprennent un trait d'union (**toto-titi**). Ces propriétés internes changent constamment au cours de la mise en page : en fait, la gravure d'une page n'est autre que le calcul de toutes ces propriétés au moyen de fonctions de rappel.

Nous pouvons donc par exemple accroître l'épaisseur des hampes en jouant sur la propriété **thickness** de l'objet **stem** :

```
c4 c
\override Voice.Stem #'thickness = #3.0
c4 c
```



Lorsqu'aucun contexte n'est spécifié dans une clause `\override`, celle-ci s'appliquera au contexte le plus bas :

```
{ \override Staff.Stem #'thickness = #3.0
  <<
    {
      e4 e
      \override Stem #'thickness = #0.5
      e4 e
    } \ {
      c4 c c c
    }
  >>
}
```



Les effets d'un `\override` prennent fin à l'aide de l'instruction `\revert` :

```
c4
\override Voice.Stem #'thickness = #3.0
c4 c
\revert Voice.Stem #'thickness
c4
```



Les effets d'un `\override` ou d'un `\revert` s'appliquent dès l'endroit où ils apparaissent, et à tous les objets dans le contexte mentionné :

```
{
  <<
    {
      e4
      \override Staff.Stem #'thickness = #3.0
      e4 e e
    } \ {
```

```

c4 c c
\revert Staff.Stem #'thickness
c4
}
>>
}

```



L'instruction `\override` doit être précédée d'un `\once` dès lors que les effets de l'accommodement ne concernent que l'événement qui la suit directement :

```

{
  <<
  {
    \override Stem #'thickness = #3.0
    e4 e e e
  } \ {
    c4
    \once \override Stem #'thickness = #3.0
    c4 c c
  }
  >>
}

```



Voir aussi

Référence des propriétés internes : [Section “Backend” dans Référence des propriétés internes](#)

5.3.4 La commande d’affinage `\tweak`

L'utilisation d'un `\override` pour modifier les propriétés d'un objet graphique affectera toutes les instances de l'objet en question au sein du contexte, et ce dès son apparition. Il peut parfois être préférable de n'affecter qu'un seul objet en particulier plutôt que tous les objets du contexte. C'est là rôle de l'instruction `\tweak`, dont la syntaxe est :

```
\tweak objet-de-rendu #'objet-propriété valeur
```

Mention de l'*objet-de-rendu* est optionnel. La commande `\tweak` s'applique à l'objet qui apparaît immédiatement après *valeur*.

Pour une introduction à la syntaxe et l'utilisation des retouches, voir le chapitre [Section “Méthodes de retouche” dans Manuel d’initiation](#).

Lorsque plusieurs éléments de même nature surviennent au même instant, il devient impossible d'utiliser l'instruction `\override` pour n'en modifier qu'un seul individuellement, d'où l'intérêt de la commande `\tweak`. Entre autres éléments qui sont susceptibles de se produire au même instant, nous citerons :

- les têtes de notes au sein d'un accord,
- les signes d'articulation,

- les liaisons de prolongation sur des notes d'un accord,
- les crochets de nolets démarrant au même instant

Dans l'exemple suivant, l'une des têtes de note de l'accord est colorisée, et l'aspect d'une autre est changé.

```
< c
  \tweak #'color #red
  d
  g
  \tweak #'duration-log #1
  a
> 4
```



L'instruction `\tweak` permet aussi de modifier l'aspect d'une liaison :

```
c-\tweak #'thickness #5 ( d e f)
```



La commande `\tweak` ne sera pleinement fonctionnelle que si elle est directement rattachée à l'objet auquel elle doit s'appliquer alors que le fichier source est converti en flux musical. Vouloir modifier la globalité d'un accord est sans résultat dans la mesure où il ne constitue qu'un conteneur pour des événements musicaux et que tous les objets seront créés à partir d'événements appartenant à un `EventChord` (un événement d'accord) :

```
\tweak #'color #red c4
\tweak #'color #red <c e>4
<\tweak #'color #red c>4
```



La commande `\tweak` simple ne saurait servir à modifier un élément qui ne serait pas explicitement mentionné dans le fichier source. C'est notamment le cas des hampes, ligatures automatiques ou altérations, dans la mesure où elles seront ultérieurement générées et après les têtes de note (objets `NoteHead`, plutôt qu'au fil des éléments musicaux saisis.

De tels objets créés indirectement peuvent être ajustés que par une forme développée de la commande `\tweak` :

```
\tweak Stem #'color #red
\tweak Beam #'color #green c8 e
<c e \tweak Accidental #'font-size #-3 ges>4
```



La commande `\tweak` ne peut non plus servir à modifier clefs ou métriques, puisqu’elles seront inmanquablement séparées du `\tweak` par l’insertion automatique d’autres éléments requis pour spécifier le contexte.

Plusieurs commandes `\tweak` en enfilade permettent d’affecter un même élément de notation :

```
c
-\tweak #'style #'dashed-line
-\tweak #'dash-fraction #0.2
-\tweak #'thickness #3
-\tweak #'color #red
\glissando
f'
```



Vous pouvez examiner le flux musical généré par une portion d’un fichier source, y compris les éléments automatiquement insérés, en suivant les indications portées à la rubrique [Section “Affichage d’expressions musicales”](#) dans *Extension de LilyPond*. Ceci s’avère tout à fait approprié pour déterminer ce qui peut se modifier à l’aide d’un `\tweak` ou bien aider à rectifier votre source de telle sorte que le `\tweak` produise ses effets.

Voir aussi

Manuel d’initiation : [Section “Méthodes de retouche”](#) dans *Manuel d’initiation*.

Manuel d’extension : [Section “Affichage d’expressions musicales”](#) dans *Extension de LilyPond*.

Problèmes connus et avertissements

Lorsqu’il y a plusieurs liaisons de prolongation dans un accord, la commande `\tweak` ne permet de modifier les points de contrôle que pour la première rencontrée dans le fichier source.

5.3.5 `\set` ou `\override`

5.3.6 Modification de listes associatives

Certaines propriétés configurables par l’utilisateur se présentent en interne comme étant des listes associatives – les puristes diront des *alists*. Une *alist* est en fait constituée de plusieurs paires de *clés* et *valeurs* ; sa structure ressemble à :

```
'((clé1 . valeur1)
  (clé2 . valeur2)
  (clé3 . valeur3)
  ...)
```

Dans le cas où cette liste représente les propriétés d’un objet graphique ou bien l’une des variables du bloc `\paper`, chaque clé peut être modifiée individuellement sans que cela n’affecte les autres.

Par exemple, pour réduire l’espacement entre deux portées adjacentes d’un même système, on utilisera la propriété `staff-staff-spacing` qui est attachée à l’objet graphique `StaffGrouper`. Cette propriété est constituée d’une liste de quatre clés : `basic-distance`, `minimum-distance`, `padding` et `stretchability`. Ses réglages par défaut tels que mentionnés à la rubrique *Back-end* de la référence des propriétés internes – voir [Section “StaffGrouper”](#) dans *Référence des propriétés internes* – sont :

```
'((basic-distance . 9)
  (minimum-distance . 7)
  (padding . 1)
  (stretchability . 5))
```

Afin de rapprocher nos deux portées, il suffit de réduire la valeur (9) de la clé `basic-distance` au niveau de celle de la clé `minimum-distance` (7). La modification d'une seule clé individuellement peut se réaliser sous la forme d'une *déclaration imbriquée* :

```
% default space between staves
\new PianoStaff <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass c1 }
>>

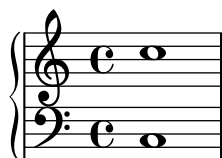
% reduced space between staves
\new PianoStaff \with {
  % this is the nested declaration
  \override StaffGrouper #'staff-staff-spacing #'basic-distance = #7
} <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass c1 }
>>
```



Le recours à une déclaration imbriquée touchera la clé indiquée (`basic-distance` dans l'exemple ci-dessus) sans pour autant modifier les autres clés de la propriété considérée.

Considérons maintenant que nous souhaitons que les portées soient le plus proche possible les unes des autres, à la limite du chevauchement. Il suffirait de mettre les quatre clés à zéro. Nous pourrions saisir quatre déclarations, chacune d'elles touchant une clé. Nous pouvons tout aussi bien redéfinir la propriété en une seule clause, sous la forme d'une liste associative :

```
\new PianoStaff \with {
  \override StaffGrouper #'staff-staff-spacing =
    #'((basic-distance . 0)
      (minimum-distance . 0)
      (padding . 0)
      (stretchability . 0))
} <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass c1 }
>>
```



N'oubliez pas que dès lors qu'une clé n'apparaît pas dans la liste, elle retourne à sa valeur *sauf-mention-contraire*. Autrement dit, dans le cas de **staff-staff-spacing** qui nous occupe, toutes les clés non mentionnées seront ramenées à zéro – à l'exception de **stretchability** qui prend par défaut la valeur de **basic-distance**. Les deux assertions suivantes sont donc équivalentes.

```
\override StaffGrouper #'staff-staff-spacing =
  #'((basic-distance . 7))
```

```
\override StaffGrouper #'staff-staff-spacing =
  #'((basic-distance . 7)
    (minimum-distance . 0)
    (padding . 0)
    (stretchability . 7))
```

L'une des conséquences, parfois involontaire, de ceci est la suppression de réglages standards effectués par un fichier d'initialisation chargé à chaque compilation d'un fichier source. Dans l'exemple précédent, les réglages standards de **padding** et **minimum-distance**, tels que déterminés par '**scm/define-grobs.scm**', se voient ramenés à leur valeur *si-non-définie* ; autrement dit, les deux clés sont mises à zéro. La définition d'une propriété ou d'une variable sous forme de liste associative, quelle qu'en soit la taille, réinitialisera toujours les clés non mentionnées à leur valeur *si-non-définie*. Si telle n'est pas votre intention, nous vous recommandons alors de régler la valeur des clés individuellement par des déclarations imbriquées.

Note : Les déclarations imbriquées ne sont pas fonctionnelles dans le cas des listes associatives des propriétés de contexte – telles **beamExceptions**, **keySignature**, **timeSignatureSettings**, etc. Ces propriétés ne sont modifiables qu'au travers d'une complète redéfinition de leur liste associative.

5.4 Propriétés et contextes utiles

5.4.1 Modes de saisie

La manière dont sera interprétée la notation contenue dans un fichier source dépend du mode affecté à la saisie.

Mode accords

Ce mode, activé par la commande **\chordmode**, permet d'interpréter les saisies comme étant des accords, qui seront imprimés sous forme de notes sur une portée – voir [Section 2.7 \[Notation des accords\]](#), page 385.

Le mode accords s'active aussi par la commande **\chords**, qui créera un contexte **ChordNames**. Les saisies, interprétées comme étant des accords, seront alors rendues sous forme nominale dans ce contexte **ChordNames** – voir [\[Impression des noms d'accord\]](#), page 391.

Mode percussions

Ce mode, activé par la commande **\drummode**, permet d'interpréter les saisies comme étant de la notation pour percussions – voir [\[Notation de base pour percussions\]](#), page 363.

Le mode percussions s'active aussi par la commande **\drums**, qui créera un contexte **DrumStaff**. Les saisies, interprétées comme étant de la notation pour percussions, seront alors rendues sous forme symbolique sur une portée de percussions – voir [\[Notation de base pour percussions\]](#), page 363.

Mode figures

Ce mode, activé par la commande `\figuremode`, permet d’interpréter les saisies comme étant de la basse chiffrée (ou figurée) – voir [Saisie de la basse chiffrée], page 400.

Le mode figures s’active aussi par la commande `\figures`, qui créera un contexte `FiguredBass`. Les saisies interprétées comme étant de la basse chiffrée, seront alors rendues sous forme symbolique dans le contexte `FiguredBass` – voir [Introduction à la basse chiffrée], page 399.

Mode frets et tablatures

Il n’existe pas de mode spécifique pour saisir des symboles de fret ou de tablature.

Notes ou accords saisis en mode note puis affectés à un contexte `TabStaff` seront rendus sous forme de diagramme de tablature – voir [Tablatures par défaut], page 319.

Deux options différentes permettent d’obtenir des diagrammes de fret en surplomb d’une portée : directement à l’aide d’un contexte `FretBoards` – voir [Tablatures automatiques], page 354 – ou en attachant aux notes des *markups* créés par la commande `\fret-diagram` – voir [Tablatures sous forme d’étiquette], page 334.

Mode paroles

Ce mode, activé par la commande `\lyricmode`, permet d’interpréter les saisies comme étant des syllabes, ayant éventuellement une durée, et des indications habituelles aux paroles – voir Section 2.1 [Musique vocale], page 238.

Le mode paroles s’active aussi par la commande `\addlyrics`, qui créera un contexte `Lyrics` et ajoutera implicitement une commande `\lyricsto` afin d’associer les paroles qui suivent à la musique précédemment saisie.

Mode markup

Ce mode, activé par la commande `\markup`, permet d’interpréter les saisies comme étant des *markups* (annotations ou étiquettes) – voir Section “Commandes pour le mode markup” dans *Manuel de notation*.

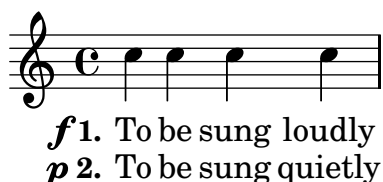
Mode notes

Le mode notes est le mode par défaut dans LilyPond. Il peut aussi s’activer par la commande `\notemode`. Les saisies seront interprétées comme étant des hauteurs, durées, *markups*, etc. qui seront rendues sous forme de notation musicale sur une portée.

Nul n’est besoin de spécifier le mode notes de manière explicite, hormis dans certaines situations particulières, notamment lorsque vous êtes en mode paroles, accords, ou tout autre mode, et que vous deviez insérer un élément qui ne serait disponible que grâce à la syntaxe du mode notes.

Il en va ainsi lorsque, par exemple, vous voulez ajouter une indication de nuance au numéro de couplet d’un chant choral :

```
{ c4 c4 c4 c4 }
\addlyrics {
  \notemode{\set stanza = \markup{ \dynamic f 1. } }
  To be sung loudly
}
\addlyrics {
  \notemode{\set stanza = \markup{ \dynamic p 2. } }
  To be sung quietly
}
```



5.4.2 Direction et positionnement

En matière de typographie musicale, l'orientation et le positionnement de nombreux éléments est affaire de goût. Par exemple, les hampes peuvent être ascendantes ou descendantes, les paroles, nuances ou autres indications d'expression peuvent apparaître au-dessus ou en dessous de la portée, les indications textuelles s'alignent tantôt par la gauche, tantôt par la droite, ou être centrées. La plupart de ces choix peuvent être laissés à l'appréciation de LilyPond. Il peut être préférable, dans certains cas, d'imposer l'orientation ou le positionnement des éléments.

Indicateurs de position d'une articulation

Certains positionnements sont opérés par défaut – toujours au-dessus ou toujours en dessous (nuances ou points d'orgue) – alors que d'autres alternent selon l'orientation des hampes (liaisons ou accents).

Le positionnement par défaut peut être outrepassé à l'aide d'un *indicateur de positionnement*, qui vient s'insérer juste avant l'articulation. LilyPond met à votre disposition trois indicateurs de positionnement : `^` (pour « au-dessus »), `_` (pour « au-dessous »), et `-` (pour « appliquer le positionnement par défaut »). L'indicateur de positionnement n'est pas obligatoire ; LilyPond considère alors qu'il y a un `-`. Un indicateur de positionnement est cependant **obligatoire** dans les cas suivants :

- une commande `\tweak`,
- une commande `\markup`,
- une commande `\tag`,
- les indications de corde, par exemple `-"corde"`,
- les indications de doigté, par exemple `-1`,
- les raccourcis d'articulation, par exemple `-. , -> ou --`.

Les indicateurs de positionnement n'affectent que la note qui suit :

```
c2( c)
c2_( c)
c2( c)
c2^( c)
```



La propriété direction

Le positionnement ou l'orientation de nombreux objets de rendu sont gérés par la propriété `direction`.

La propriété `direction` peut prendre la valeur `1`, qui signifie « ascendant » ou « au-dessus », ou `-1`, qui signifie « descendant » ou « au-dessous ». Les symboliques `UP` et `DOWN` peuvent remplacer respectivement `1` et `-1`. Les valeurs `0` ou `CENTER` permettent de réaffecter à la propriété `direction` son comportement par défaut. Certaines commandes prédéfinies permettent par ailleurs de spécifier un comportement en matière d'orientation ou positionnement ; elles sont de la forme

`\xxxUp`, `\xxxDown` et `\xxxNeutral`

auquel cas `\xxxNeutral` signifie « retour au comportement par défaut » – voir [Section “Objets inclus dans la portée”](#) dans *Manuel d'initiation*.

Dans quelques cas particuliers, comme l'indication d'un *arpeggio*, la valeur affectée à la propriété **direction** déterminera si l'objet doit se placer à gauche ou à droite de son parent. Un **-1** ou **LEFT** signifiera alors « sur la gauche », et un **1** ou **RIGHT** « sur la droite ». Comme de bien entendu, un **0** ou **CENTER** signifiera « appliquer le positionnement par défaut ».

Notez que ces commandes resteront effectives jusqu'à ce qu'elles soient annulées.

```
c2( c)
\slurDown
c2( c)
c2( c)
\slurNeutral
c2( c)
```



En matière de musique polyphonique, il est souvent plus judicieux d'utiliser des contextes **Voice** explicites que de modifier l'orientation des objets. Pour de plus amples informations, voir [Section 1.5.2 \[Plusieurs voix\]](#), page 154.

Voir aussi

Manuel d'initiation : [Section “Objets inclus dans la portée”](#) dans *Manuel d'initiation*.

Manuel de notation : [Section 1.5.2 \[Plusieurs voix\]](#), page 154.

5.4.3 Distances et unités de mesure

LilyPond considère deux types de distances : les distances absolues et les distances relatives ou extensibles.

Les distances absolues permettent de spécifier les marges, indentations et autres détails de mise en page ; elles s'expriment par défaut en millimètres. Vous pouvez utiliser d'autres systèmes de mesure, dès lors que la quantité est suivie de la mesure : `\mm`, `\cm`, `\in` (pouces) ou `\pt` (points, 1/72,27 pouce). Les mesures de mise en page peuvent aussi s'exprimer en unité extensible de portée `\staff-space` (voir ci-après). Pour plus d'information concernant la mise en page, voir la rubrique [Section 4.1 \[Mise en forme de la page\]](#), page 496.

Les distances relatives ou extensibles s'expriment toujours en « espace de portée » ou, plus rarement, en « demi espace de portée ». L'espace de portée correspond à la distance qui sépare deux lignes adjacentes d'une portée. Sa valeur par défaut est déterminée globalement par la taille de portée. Elle peut aussi s'ajuster ponctuellement en jouant sur la propriété **staff-space** de l'objet **StaffSymbol**. Les distances relatives s'ajustent automatiquement dès qu'une modification de la taille globale de portée ou bien de la propriété **staff-space** du **StaffSymbol** intervient. Cependant, les tailles de fonte ne s'ajusteront automatiquement que si la modification touche la taille globale des portées. La taille globale de portée permet ainsi de gérer l'aspect général de la partition – voir [Section 4.2.2 \[Définition de la taille de portée\]](#), page 509.

Lorsque seulement une portion de partition doit apparaître dans une taille, comme par exemple une portée d'ossia ou une note de bas de page, influencer sur la taille globale de portée affecterait l'intégralité de la partition. Il convient donc dans ce cas de modifier à la fois la propriété **staff-space** du **StaffSymbol** et la taille des fontes. La fonction Scheme **magstep** est tout spécialement chargée d'adapter une modification du **staff-space** aux fontes. Pour de plus amples informations, reportez-vous à la rubrique [Section “Longueur et épaisseur des objets”](#) dans *Manuel d'initiation*.

Voir aussi

Manuel d'initiation : [Section “Longueur et épaisseur des objets”](#) dans *Manuel d'initiation*.

Manuel de notation : [Section 4.2.2 \[Définition de la taille de portée\]](#), page 509, [Section 4.1 \[Mise en forme de la page\]](#), page 496.

5.4.4 Propriétés des symboles de la portée

L'emplacement vertical et le nombre de lignes d'une portée se définissent conjointement. Comme l'illustre l'exemple suivant, le positionnement des notes n'est en rien influencé par le positionnement des lignes de la portée.

Note : La propriété `'line-positions` écrase la propriété `'line-count`. Le nombre de lignes d'une portée est implicitement défini par le nombre d'éléments dans la liste des valeurs de `'line-positions`.

```
\new Staff \with {
  \override StaffSymbol #'line-positions = #'(7 3 0 -4 -6 -7)
}
{ a4 e' f b | d1 }
```



La largeur d'une portée, exprimée en espace de portée, peut être figée. L'espacement des objets inclus dans cette portée ne sera en rien affecté par ce réglage.

```
\new Staff \with {
  \override StaffSymbol #'width = #23
}
{ a4 e' f b | d1 }
```



5.4.5 Extenseurs et prolongateurs

De nombreux objets de notation musicale s'étendent sur plusieurs notes, voire même sur plusieurs mesures. Il en va ainsi des liaisons, ligatures, crochets de nolet, crochets de reprise, crescendos, trilles ou glissandos. Ces objets, que l'on englobe sous l'appellation « d'extenseur », sont pourvus de propriétés spécifiques destinées à contrôler leur apparence et leur comportement. Un certain nombre de ces propriétés sont communes à tous les extenseurs, d'autres n'affectent que certains d'entre eux.

Tout extenseur dispose de la `spanner-interface`. Quelques uns, tout particulièrement ceux chargés de dessiner une ligne droite entre deux objets, disposent aussi de la `line-spanner-interface`.

Utilisation de `spanner-interface`

Cette interface fournit deux propriétés qui s'appliquent à certains extenseurs.

La propriété `minimum-length`

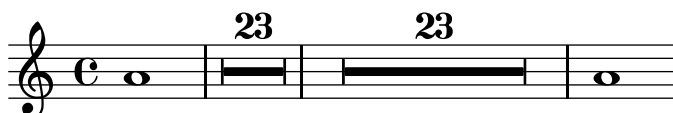
La longueur minimale d'un extenseur est déterminée par la propriété `minimum-length`. Au plus sa valeur est élevée, au plus l'espacement des notes qui le bornent sera grand. Forcer sa

valeur restera néanmoins sans effet pour un certain nombre d'extenseurs dont la longueur dépend d'autres considérations. Voici quelques exemples de mise en œuvre de cette propriété.

```
a~a
a
% increase the length of the tie
-\tweak #'minimum-length #5
~a
```



```
a1
\compressFullBarRests
R1*23
% increase the length of the rest bar
\once \override MultiMeasureRest #'minimum-length = #20
R1*23
a1
```



```
a \< a a a \!
% increase the length of the hairpin
\override Hairpin #'minimum-length = #20
a \< a a a \!
```



Cette propriété permet aussi de jouer sur l'envergure d'une liaison d'articulation ou de phrasé.

```
a( a)
a
-\tweak #'minimum-length #5
( a)
```

```
a\ ( a\ )
a
-\tweak #'minimum-length #5
\ ( a\ )
```



Certains objets requièrent un appel explicite à la procédure `set-spacing-rods` pour que la propriété `minimum-length` produise ses effets. La propriété `set-spacing-rods` doit alors prendre pour valeur `ly:spanner::set-spacing-rods`. Par exemple, la longueur minimale d'un glissando ne pourra être forcée tant que la propriété `springs-and-rods` n'est pas définie :


```
% default
e \glissando c'

% not effective alone
\once \override Glissando #'minimum-length = #20
e, \glissando c'

% effective only when both overrides are present
\once \override Glissando #'minimum-length = #20
\once \override Glissando #'springs-and-rods = #ly:spanner::set-spacing-rods
e, \glissando c'
```



Il en va de même pour l'objet `Beam` (ligature) :

```
% not effective alone
\once \override Beam #'minimum-length = #20
e8 e e e

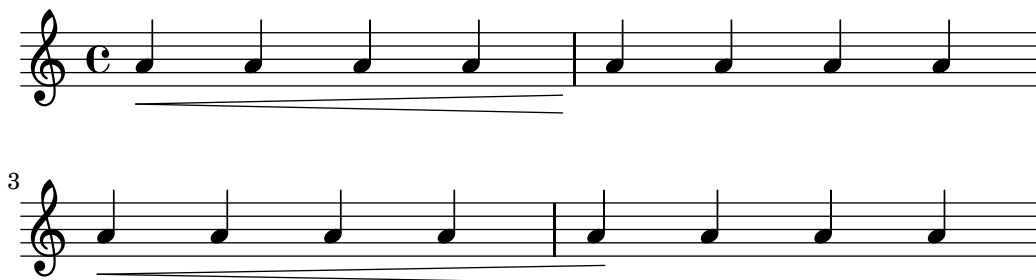
% effective only when both overrides are present
\once \override Beam #'minimum-length = #20
\once \override Beam #'springs-and-rods = #ly:spanner::set-spacing-rods
e8 e e e
```



La propriété `to-barline`

La seconde propriété fournie par la `spanner-interface` est `to-barline`. Elle est activée par défaut, raison pour laquelle les soufflets et autres extenseurs finissant sur la première note d'une mesure s'arrêtent visuellement au niveau de la barre de mesure qui la précède. Le fait de désactiver la propriété `to-barline` aura pour effet de prolonger l'extenseur au delà de la barre de mesure et jusqu'à la note qui le borne :

```
a \< a a a a \! a a a \break
\override Hairpin #'to-barline = ##f
a \< a a a a \! a a a
```



Cette propriété n'est pas opérationnelle pour tous les extenseurs. Il serait en effet quelque peu surprenant de l'activer (lui affecter `#t`) dans le cas d'une liaison d'articulation ou de phrasé !

Utilisation de `line-spanner-interface`

Un certain nombre d'objets disposent de la propriété `line-spanner-interface`, entre autres :

- `DynamicTextSpanner`
- `Glissando`
- `TextSpanner`
- `TrillSpanner`
- `VoiceFollower`

La routine en charge de dessiner le stencil de ces extenseurs est `ly:line-interface::print`. Elle va déterminer les deux points extrêmes et dessiner entre eux une ligne du style requis. Bien que la localisation des deux bornes de l'extenseur soit calculée à la volée, vous pouvez cependant forcer leur ordonnée (coordonnée-Y). Les propriétés que vous devrez ajuster résident au deuxième niveau dans la hiérarchie, mais la syntaxe de la commande `\override` nécessaire demeure relativement simple :

```
e2 \glissando b
\once \override Glissando #'(bound-details left Y) = #3
\once \override Glissando #'(bound-details right Y) = #-2
e2 \glissando b
```



La propriété `Y` est valorisée en unités de `staff-space`, la ligne médiane de la portée correspondant au point zéro. Pour le glissando qui nous occupe, il s'agit du `Y` à l'aplomb (coordonnée-X) du centre de la tête de chacune des deux notes, si tant est que la ligne doive s'étendre entre ces deux points.

Si le `Y` n'est pas défini, sa valeur sera calculée en fonction de la position verticale du point d'attachement de l'extenseur.

Dans le cas où l'extenseur est interrompu par un saut de ligne, les terminaisons à cet endroit se gèrent grâce aux sous-clés `left-broken` et `right-broken` de `bound-details`, comme ci-dessous :

```
\override Glissando #'breakable = ##t
\override Glissando #'(bound-details right-broken Y) = #-3
c1 \glissando \break
f1
```



Les sous-propriétés `left` et `right` du `bound-details` disposent d'autres clés modifiables de la même manière que `Y` :

- | | |
|----------------|--|
| <code>Y</code> | Détermine l'ordonnée (coordonnée-Y) de la terminaison, avec un décalage en <code>staff-space</code> par rapport à la ligne médiane de la portée. Il s'agit par défaut du centre de l'objet d'attachement, qui est le centre vertical de la tête de note pour un glissando. En ce qui concerne les extenseurs horizontaux, tels ceux comportant du texte ou les trilles, il est fixé à 0. |
|----------------|--|

attach-dir

Détermine le début et la fin de la ligne sur l'axe des abscisses, relativement à l'objet de rattachement. Une valeur de -1 (ou **LEFT**) aura pour effet de commencer ou terminer la ligne sur la gauche de la tête de note de rattachement.

X

Il s'agit de l'abscisse (coordonnée-X) absolue de la terminaison. Elle se calcule à la volée, et son forçage n'apporte rien de plus.

stencil

Les extenseurs linéaires peuvent commencer ou finir par un symbole, enregistré dans cette sous-propriété. Elle est conçue pour un usage interne, aussi nous vous conseillons de plutôt recourir à **text**.

text

Il s'agit d'un *markup* qui se poursuivra par l'extenseur. C'est la sous-propriété utilisée pour ajouter *cresc.*, *tr* ou autre texte à un extenseur horizontal.

```
\override TextSpanner #'(bound-details left text)
    = \markup { \small \bold Slower }
c2\startTextSpan b c a\stopTextSpan
```

**stencil-align-dir-y****stencil-offset**

Lorsqu'aucune de ces deux sous-proPERTIES n'est définie, le stencil est simplement positionné à l'extrémité, centré sur la ligne telle que définie par les sous-proPERTIES X et Y. L'utilisation de **stencil-align-dir-y** ou **stencil-offset** permettra d'aligner le symbole verticalement par rapport au coin de la ligne :

```
\override TextSpanner
    #'(bound-details left stencil-align-dir-y) = #-2
\override TextSpanner
    #'(bound-details right stencil-align-dir-y) = #UP

\override TextSpanner
    #'(bound-details left text) = #"ggg"
\override TextSpanner
    #'(bound-details right text) = #"hhh"
c4^\startTextSpan c c c \stopTextSpan
```



Vous n'aurez pas manqué de constater qu'une valeur négative place le texte *en haut* – contrairement à ce que l'on serait en droit d'attendre. Ceci est dû au fait que la valeur -1 ou **DOWN** signifie « aligner le bord *inférieur* du texte sur la ligne d'extension ». Une valeur égale à 1 ou **UP** alignera le sommet du texte sur cette ligne d'extension.

arrow

L'activation de cette sous-propriété (lui affecter **#t**) ajoutera à l'extenseur une terminaison en flèche.

padding Cette sous-propriété contrôle l'espace qui doit séparer l'extrémité de la ligne et la fin réelle de l'extenseur. Sans ce « décalage », le trait indiquant un glissando commencerait et finirait au beau milieu de chacune des têtes de note.

La fonction `\endSpanners` permet d'interrompre l'extenseur qui vient dès la note suivante. Autrement dit, il ne s'étendra que sur une seule note, ou jusqu'à la prochaine barre de mesure si `to-barline` a été activé et que survient une barre avant la note suivante.

```
\endSpanners
c2 \startTextSpan c2 c2
\endSpanners
c2 \< c2 c2
```



L'utilisation de `\endSpanners` permet de s'affranchir d'insérer un `\stopTextSpan` pour clôturer un `\startTextSpan` ou un `\!` pour terminer un soufflet.

Voir aussi

Référence des propriétés internes : Section “Glissando” dans *Référence des propriétés internes*, Section “line-spanner-interface” dans *Référence des propriétés internes*, Section “TextSpanner” dans *Référence des propriétés internes*, Section “TrillSpanner” dans *Référence des propriétés internes*, Section “VoiceFollower” dans *Référence des propriétés internes*.

5.4.6 Visibilité des objets

La visibilité des objets de rendu se contrôle de quatre façons différentes : vous pouvez supprimer leur stencil, les rendre transparents, les coloriser en blanc ou bien encore forcer leur propriété `break-visibility`. Les trois premières options peuvent s'appliquer à tous les objets, la dernière étant réservée aux objets *changeables*. Le Manuel d'initiation contient une introduction à ces quatre techniques, à la rubrique Section “Visibilité et couleur des objets” dans *Manuel d'initiation*.

LilyPond met en œuvre quelques techniques particulières adaptées à certains objets ; elles sont couvertes par une rubrique spécifique.

Suppression des stencils

Tout objet de rendu se voit attribuer une propriété `stencil`. Elle est par défaut définie par la fonction chargée de dessiner cet objet. Lorsque cette propriété est désactivée de force – en lui attribuant la valeur `#f` – aucune fonction ne sera appelée ; l'objet ne sera donc pas dessiné. Le retour au comportement par défaut s'opère à l'aide d'un `\revert`.

```
a1 a
\override Score.BarLine #'stencil = ##f
a a
\revert Score.BarLine #'stencil
a a a
```



Transparence des objets

Tout objet de rendu dispose d'une propriété de transparence, qui est par défaut définie à **#f**. Le fait de l'activer rendra l'objet transparent tout en préservant la place qu'il occupe.

```
a4 a
\once \override NoteHead #'transparent = ##t
a a
```



Blanchiment des objets

Tout objet de rendu dispose d'une propriété couleur, par défaut définie à **black** (noir). Le fait de la forcer à **white** (blanc) rendra l'objet indistinct du fond blanc. Néanmoins, lorsque cet objet en recouvre d'autres, la couleur de leurs points de jonction dépendra de l'ordre dans lequel ils sont dessinés, ce qui peut laisser apparaître un fantôme de l'objet blanchi comme ci-dessous :

```
\override Staff.Clef #'color = #white
a1
```



Cet inconvénient peut être évité en modifiant l'ordre dans lequel les objets sont dessinés. Chaque objet de rendu dispose d'une propriété **layer** (calque ou niveau) à laquelle est affecté un nombre entier. Les objets ayant la plus faible valeur sont dessinés en premier, puis les autres, de telle sorte qu'un objet ayant une valeur plus élevée les recouvrira. La plupart des objets ont un **layer** valorisé à 1 – quelques uns, dont **StaffSymbol** et **BarLine**, ont une valeur à 0. L'ordre d'impression d'objets ayant une même valeur de **layer** est indéterminé.

La clef de l'exemple précédent a par défaut un **layer** à 1 ; elle est donc dessinée après les lignes de la portée – **layer** valorisé par défaut à 0 – et donc les recouvre. Pour changer cet état de fait, l'objet **Clef** doit avoir un **layer** de valeur inférieure, disons -1, pour pouvoir être dessiné avant.

```
\override Staff.Clef #'color = #white
\override Staff.Clef #'layer = #-1
a1
```



Utilisation de break-visibility

La plupart des objets de rendu ne sont imprimés qu'une seule fois ; certains cependant, tels les barres de mesure, clefs, métriques ou armures, apparaîtront deux fois lors d'un saut de ligne – une première fois en fin de ligne, puis à nouveau au début de la ligne suivante. Ces objets, que l'on peut traiter de *changeables* (*breakable* en anglais) disposent de la propriété **break-visibility** spécialement chargée de contrôler leur visibilité aux trois endroits où il sont susceptibles d'apparaître : en début de ligne, en cours de ligne ou en fin de ligne – si tant est qu'un changement s'y produise.

Par exemple, la métrique est imprimée par défaut au début de la première ligne, et nulle part ailleurs. En cas de modification, une nouvelle métrique sera imprimée à l'endroit du changement.

Dans le cas où ce changement intervient en fin de ligne, la nouvelle métrique s'imprime au début de la ligne suivante, et une métrique « de précaution » viendra se placer au bout de la ligne précédente.

Ce comportement est géré par la propriété **break-visibility**, dont vous trouverez une explication à la rubrique **Section “Visibilité et couleur des objets”** dans *Manuel d'initiation*. Cette propriété est constituée d'un vecteur de trois booléens qui, dans l'ordre, déterminent si l'objet sera imprimé à la fin, en cours, et au début d'une ligne – on pourrait aussi dire avant un saut de ligne, là où il n'y a pas de saut de ligne, et après un saut de ligne.

Les huit combinaisons possibles sont aussi disponibles sous la forme de fonctions prédéfinies, regroupées dans le fichier `'scm/output-lib.scm'`. Le tableau suivant vous les présente ; les trois dernières colonnes indiquent l'endroit où l'objet sera visible.

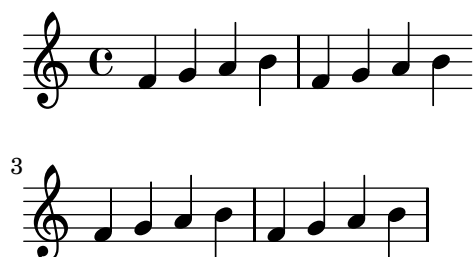
Forme fonctionnelle	Forme vectorielle	Avant saut	Hors saut	Après saut
<code>all-visible</code>	<code>'#(#t #t #t)</code>	oui	oui	oui
<code>begin-of-line-visible</code>	<code>'#(#f #f #t)</code>	non	non	oui
<code>center-visible</code>	<code>'#(#f #t #f)</code>	non	oui	non
<code>end-of-line-visible</code>	<code>'#(#t #f #f)</code>	oui	non	non
<code>begin-of-line-invisible</code>	<code>'#(#t #t #f)</code>	oui	oui	non
<code>center-invisible</code>	<code>'#(#t #f #t)</code>	oui	non	oui
<code>end-of-line-invisible</code>	<code>'#(#f #t #t)</code>	non	oui	oui
<code>all-invisible</code>	<code>'#(#f #f #f)</code>	non	non	non

Les réglages par défaut de la propriété **break-visibility** diffèrent selon l'objet de rendu. Le tableau suivant présente, pour la plupart des objets comportant la propriété **break-visibility**, ces réglages par défaut.

Objet de rendu	Contexte habituel	Réglage par défaut
<code>BarLine</code>	Score	calculé
<code>BarNumber</code>	Score	<code>begin-of-line-visible</code>
<code>BreathingSign</code>	Voice	<code>begin-of-line-invisible</code>
<code>Clef</code>	Staff	<code>begin-of-line-visible</code>
<code>Custos</code>	Staff	<code>end-of-line-visible</code>
<code>DoublePercentRepeat</code>	Voice	<code>begin-of-line-invisible</code>
<code>KeySignature</code>	Staff	<code>begin-of-line-visible</code>
<code>OctavateEight</code>	Staff	<code>begin-of-line-visible</code>
<code>RehearsalMark</code>	Score	<code>end-of-line-invisible</code>
<code>TimeSignature</code>	Staff	<code>all-visible</code>

Voici un exemple d'utilisation de la forme vectorielle pour contrôler la visibilité des barres de mesure :

```
f4 g a b
f4 g a b
% Remove bar line at the end of the current line
\once \override Score.BarLine #'break-visibility = #'(#f #t #t)
\break
f4 g a b
f4 g a b
```



Lors d'un forçage de **break-visibility** sous une forme vectorielle, les trois éléments doivent impérativement être mentionnés. Ces formes vectorielles ne sont d'ailleurs pas prises en charge par tous les objets de rendu, et certaines combinaisons peuvent entraîner des erreurs ; nous citerons entre autres :

- Une barre de mesure ne peut s'imprimer en début de ligne.
- Un numéro de mesure ne peut apparaître au début de la première ligne, à moins d'être différent de 1.
- Clef – voir ci-après.
- Les répétitions en pourcentage sont soit toutes imprimées, soit aucune. Vous devrez utiliser **begin-of-line-invisible** pour les voir et **all-invisible** pour les masquer.
- Armure – voir ci-après.
- Indication d'octavation – voir ci-après.

Considérations spécifiques

Visibilité après changement explicite

La propriété **break-visibility** contrôle la visibilité des armures ou changements de clef en début de ligne uniquement, donc après un saut. Elle ne produit aucun effet sur la visibilité d'une armure ou d'une clef après un changement explicite de tonalité ou de clef, ni en cours, ni en fin de ligne. Dans l'exemple suivant, l'armure est présente même après le passage en si bémol majeur malgré l'activation de **all-invisible** (*tous invisibles*).

```
\key g \major
f4 g a b
% Try to remove all key signatures
\override Staff.KeySignature #'break-visibility = #all-invisible
\key bes \major
f4 g a b
\break
f4 g a b
f4 g a b
```



La visibilité lors de ces changements explicites d'armure ou de clef est géré respectivement par les propriétés **explicitKeySignatureVisibility** et **explicitClefVisibility**. Leur fonctionnement est en tout point identique à celui de la propriété **break-visibility** – forme vectorielle à trois éléments ou forme fonctionnelle comme indiqué ci-avant. Toutes deux sont attachées au contexte **Staff** (la portée) et non directement aux objets de rendu ; elles sont de ce fait introduites par une instruction **\set**. Leur valeur par défaut est de toujours imprimer les objets

– réglage sur `all-visible`. Ces deux propriétés gèrent uniquement la visibilité des armures et clefs lors d'un changement explicite, et en dehors d'un début de ligne ; il faudra en pareil cas forcer la `break-visibility` de ces objets pour les supprimer.

```
\key g \major
f4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Staff.KeySignature #'break-visibility = #all-invisible
\key bes \major
f4 g a b \break
f4 g a b
f4 g a b
```



Visibilité des altérations de précaution

L'impression d'altérations de précaution au moment d'un changement explicite de tonalité sera annulée dès lors que vous aurez désactivé la propriété `printKeyCancellation` du contexte `Staff` :

```
\key g \major
f4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.printKeyCancellation = ##f
\override Staff.KeySignature #'break-visibility = #all-invisible
\key bes \major
f4 g a b \break
f4 g a b
f4 g a b
```



Avec de tels réglages particuliers, seules les altérations accidentelles permettront d'indiquer le changement de tonalité.

Barres de mesure automatiques

La désactivation de la propriété `automaticBars`, qui réside dans le contexte `Score`, permet de s'affranchir d'imprimer automatiquement les barres de mesure ; seules seront imprimées les barres explicitées à l'aide de la commande `\bar`. Néanmoins, et contrairement à ce qui se passe avec la commande `\cadenzaOn`, le compteur de numéro de mesure continuera de s'incrémenter. Les barres s'imprimeront à nouveau, au niveau où en est le compteur, dès que la propriété

`automaticBars` sera réactivée. Gardez à l'esprit que les sauts de ligne, lorsque cette propriété est désactivée, ne peuvent intervenir qu'à l'occasion d'un `\bar` explicite.

Clefs octaviées

L'indication d'octaviation d'une clef est produite par l'objet de rendu `OctavateEight`. Sa visibilité étant gérée par héritage direct de l'objet `Clef`, nul n'est besoin de forcer un quelconque `break-visibility` au niveau des objets `OctavateEight` pour éliminer une indication d'octaviation lorsque la clef est invisible.

Lors d'un changement explicite de clef, la propriété `explicitClefVisibility` gère à la fois le symbole de la clef et l'indication d'octaviation qui lui est attachée.

Voir aussi

Manuel d'initiation : [Section “Visibilité et couleur des objets”](#) dans *Manuel d'initiation*

5.4.7 Styles de ligne

Certaines indications portées à l'attention de l'exécutant – tels *rallentando*, *accelerando* et *trilles* – apparaissent sous la forme d'un texte qui peut s'étendre sur plusieurs mesures à l'aide d'une ligne parfois pointillée ou ondulée.

En matière de dessin du texte et des lignes, ces indications font appel aux mêmes routines que le glissando ; leur comportement peut donc être affiné selon les mêmes préceptes, au moyen de la routine `ly:line-interface::print` qui est tout spécialement chargée de dessiner les extenseurs. Cette routine détermine l'emplacement exact des deux points extrêmes de l'extenseur, puis trace une ligne du style demandé entre ces deux points.

L'exemple ci-dessous indique les différents styles de ligne disponibles, ainsi que la manière de les spécifier.

```
d2 \glissando d'2
\once \override Glissando #'style = #'dashed-line
d,2 \glissando d'2
\override Glissando #'style = #'dotted-line
d,2 \glissando d'2
\override Glissando #'style = #'zigzag
d,2 \glissando d'2
\override Glissando #'style = #'trill
d,2 \glissando d'2
```



Les points d'ancrage de l'extension sont calculés à la volée pour chaque objet graphique, mais rien ne vous empêche de les forcer :

```
e2 \glissando f
\once \override Glissando #'(bound-details right Y) = #-2
e2 \glissando f
```



La valeur de `Y` est ainsi fixée à `-2` en ce qui concerne la borne droite. Il en irait de même pour la borne gauche en spécifiant `left` (gauche) au lieu de `right` (droite).

En l'absence de réglage du Y, celui-ci est calculé à partir de l'emplacement vertical des points d'attache gauche et droit de l'extenseur.

De plus amples informations quant à l'ajustement des extenseurs font l'objet de la rubrique [Section 5.4.5 \[Extenseurs et prolongateurs\]](#), page 581.

5.4.8 Rotation des objets

Qu'il s'agisse des objets de rendu ou d'éléments textuels sous forme de *markup*, vous pouvez les faire pivoter selon vos désirs et à partir de n'importe quel point. La méthode diffère cependant selon ce que vous désirez manipuler.

Rotation des objets de mise en forme

Tout objet de rendu disposant de la **grob-interface** est susceptible de pivoter, grâce à la propriété **rotation**. Celle-ci prend en argument une liste de trois éléments : l'angle de rotation – dans le sens inverse des aiguilles d'une montre – ainsi que les coordonnées **x** et **y** du point appartenant à l'objet en question et à partir duquel doit s'effectuer cette rotation. L'angle est exprimé en degrés, les coordonnées en espaces de portée.

L'angle et les coordonnées ne peuvent se déterminer que par tâtonnement.

Il existe assez peu de situation où faire pivoter un objet de mise en forme soit réellement opportun ; en voici une :

```
g4\< e' d' f\!  
\override Hairpin #'rotation = #'(20 -1 0)  
g,,4\< e' d' f\!
```



Rotation des étiquettes

Tout texte faisant l'objet d'un *markup* peut pivoter selon n'importe quel angle, à l'aide de la commande **\rotate**. Celle-ci prend deux arguments : l'angle de rotation exprimé en degrés – dans le sens inverse des aiguilles d'une montre – et le texte à basculer. Il ne s'agit pas ici de faire pivoter les extrémités du texte ; celles-ci récupéreront leurs coordonnées **x** et **y** du *markup* pivoté. Dans l'exemple ci-dessous, la propriété **outside-staff-priority** a été fixée à **##f** afin de désactiver l'évitement automatique des collisions qui pourrait repousser certains textes trop haut.

```
\override TextScript #'outside-staff-priority = ##f  
g4^\markup { \rotate #30 "un sol" }  
b^\markup { \rotate #30 "un si" }  
des^\markup { \rotate #30 "un ré bémol" }  
fis^\markup { \rotate #30 "un fa dièse" }
```



5.5 Retouches avancées

Nous allons voir, au fil des paragraphes qui suivent, différentes approches permettant de figurer l'apparence d'une partition.

Voir aussi

Manuel d'initiation : Section “Autres sources de documentation” dans *Manuel d'initiation*, Section “Retouche de partition” dans *Manuel d'initiation*.

Manuel de notation : Section 5.2 [En quoi consiste la référence des propriétés internes], page 565, Section 5.3 [Modification de propriétés], page 569.

Manuel d'extension : Section “Interfaces pour programmeurs” dans *Extension de LilyPond*.

Fichiers d'initialisation : ‘scm/define-grobs.scm’.

Morceaux choisis : Section “Retouches” dans *Morceaux choisis*.

Référence des propriétés internes : Section “All layout objects” dans *Référence des propriétés internes*.

5.5.1 Alignement des objets

Les objets graphiques disposant des interfaces **self-alignment-interface** ou **side-position-interface** peuvent s'aligner par rapport à un objet précédemment positionné, ce de différentes manières. Ces objets sont référencés aux rubriques Section “self-alignment-interface” dans *Référence des propriétés internes* et Section “side-position-interface” dans *Référence des propriétés internes*.

Tous les objets graphiques ont un point de référence, une étendue horizontale et une étendue verticale. L'étendue horizontale est représentée par une paire de nombres indiquant l'écart entre le point de référence et les bords gauche et droit – l'écart à gauche étant négatif. L'étendue verticale est représentée par une paire de nombres indiquant l'écart entre le point de référence et les bords inférieur et supérieur – l'écart vers le bas étant négatif.

La position d'un objet sur la portée est donnée par la valeur des propriétés **X-offset** et **Y-offset**. La valeur de **X-offset** indique l'écart en abscisse (coordonnée X) par rapport au point de référence de l'objet parent ; la valeur de **Y-offset** indique l'écart par rapport à la ligne médiane de la portée. Les valeurs de **X-offset** et **Y-offset** peuvent être fournies arbitrairement, ou bien être calculé par des procédures spécifiques qui détermineront l'alignement par rapport à l'objet parent.

Note : Nombre d'objets sont affectés par des considérations spécifiques en matière de positionnement ; jouer sur les valeurs de **X-offset** ou **Y-offset** se révélera inefficace en pareil cas, même si l'objet dispose de la **self-alignment-interface**. Fixer arbitrairement les propriétés **X-offset** ou **Y-offset** annihilerait alors les effets de la propriété **self-alignment** correspondante.

Par exemple, une altération peut se repositionner verticalement grâce à son **Y-offset** ; toute modification de son **X-offset** restera par contre sans effet.

Les indications de repère s'alignent sur des objets de rupture – tels les barres de mesure, clefs, métriques et armures. Certaines propriétés spécifiques – dépendant de la **break-aligned-interface** – permettent de gérer le positionnement des indications de repère sur ces objets.

Voir aussi

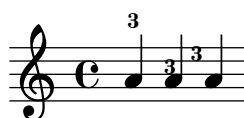
Manuel de notation : [Utilisation de **break-aligned-interface**], page 596.

Manuel d'extension : Section “Fonctions de rappel” dans *Extension de LilyPond*.

Détermination directe de X-offset et Y-offset

Vous pouvez fournir, pour de nombreux objets, des valeurs numériques aux propriétés **X-offset** et **Y-offset**. Voici par exemple une note avec indication du doigté tout d'abord avec un positionnement par défaut, puis repositionnement par modification successive du **X-offset** et du **Y-offset**.

```
a-3
a
-\tweak #'X-offset #0
-\tweak #'Y-offset #0
-3
a
-\tweak #'X-offset #-1
-\tweak #'Y-offset #1
-3
```



Utilisation de side-position-interface

Un objet disposant de la **side-position-interface** peut se voir accolé à son voisin de telle sorte que les bords des deux objets se touchent. Un tel objet peut se positionner au-dessus, en dessous, à droite ou à gauche de son parent. Ce parent ne saurait être stipulé ; il est déterminé par l'ordre d'apparition des éléments dans le flux des saisies. La plupart de ces objets ont pour parent une tête de note.

Les valeurs des propriétés **side-axis** et **direction** détermineront l'endroit où viendra se positionner l'objet, selon les préceptes suivants :

Propriété side-axis	Propriété direction	Positionnement
0	-1	gauche
0	1	droite
1	-1	en dessous
1	1	au-dessus

Pour un **side-axis** à 0, le **X-offset** devrait engager la procédure **ly:side-position-interface::x-aligned-side**. Celle-ci renverra la valeur adéquate de **X-offset** permettant d'accoler l'objet sur la droite ou sur la gauche de son parent, selon la valeur de **direction**.

Pour un **side-axis** à 1, le **Y-offset** devrait engager la procédure **ly:side-position-interface::y-aligned-side**. Celle-ci renverra la valeur adéquate de **Y-offset** permettant d'accoler l'objet au-dessus ou en dessous de son parent, selon la valeur de **direction**.

Utilisation de self-alignment-interface

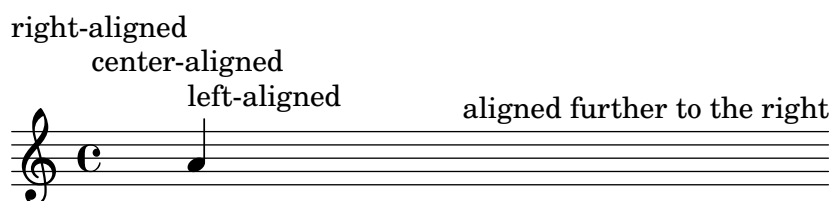
Réalignement d'objets horizontalement

L'alignement horizontal d'un objet disposant de la **self-alignment-interface** dépend de la valeur de sa propriété **self-alignment-X**, si tant est que la propriété **X-offset** de cet objet engage la procédure **ly:self-alignment-interface::x-aligned-on-self**. La propriété **self-alignment-X** peut contenir un nombre réel, l'unité de base étant la moitié de l'étendue horizontale de l'objet. Une valeur négative décalera l'objet vers la droite, une valeur positive vers la gauche. La valeur 0 permet de centrer l'objet sur le point de référence de son parent. Une

valeur de -1 alignera le bord gauche de l'objet sur le point de référence de son parent, et une valeur de 1 alignera le bord droit de l'objet sur le point de référence de son parent. Les valeurs symboliques LEFT, CENTER et RIGHT correspondent respectivement à -1, 0 et 1.

En règle générale, la valeur de `self-alignment-X` se modifie à l'aide d'une commande `\override`. Le recours à la commande `\tweak` permet de traiter séparément plusieurs annotations affectées à une même note :

```
a'
-\tweak #'self-alignment-X #-1
^"left-aligned"
-\tweak #'self-alignment-X #0
^"center-aligned"
-\tweak #'self-alignment-X #RIGHT
^"right-aligned"
-\tweak #'self-alignment-X #-2.5
^"aligned further to the right"
```



Réalignement d'objets verticalement

L'alignement vertical suit le même principe : la propriété `Y-offset` doit alors engager la procédure `ly:self-alignment-interface::y-aligned-on-self`. Toutefois, il arrive bien souvent que d'autres mécanismes interviennent dans l'alignement vertical. La valeur de `Y-offset` n'étant que l'une des variables qui seront prises en compte, l'ajustement pour certains objets peut se révéler fastidieux. L'unité de base est relativement réduite, puisqu'elle est de la moitié de l'étendue verticale de l'objet ; le nombre à fournir en argument pourrait donc être relativement élevé. Une valeur de -1 alignera le bord inférieur de l'objet sur le point de référence de son parent, et une valeur de 1 alignera le bord supérieur de l'objet sur le point de référence de son parent. La valeur 0 permet de centrer l'objet sur le point de référence de son parent. Les valeurs symboliques DOWN, CENTER et UP correspondent respectivement à -1, 0 et 1.

Réalignement d'objets sur les deux axes

Définir à la fois `X-offset` et `Y-offset` permet de réaligner un objet sur les deux axes.

Dans l'exemple ci-dessous, nous ajustons l'indication de doigté de telle sorte qu'elle se place au plus près de la tête de note.

```
a
-\tweak #'self-alignment-X #0.5 % move horizontally left
-\tweak #'Y-offset #ly:self-alignment-interface::y-aligned-on-self
-\tweak #'self-alignment-Y #-1 % move vertically up
-3 % third finger
```

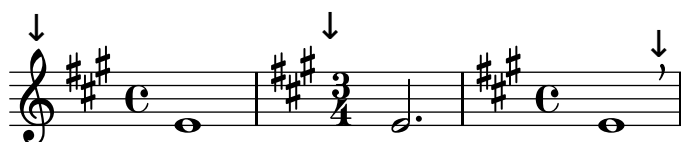


Utilisation de break-aligned-interface

Indications de repère et numéros de mesure peuvent s'aligner sur des objets de notation autres qu'une barre de mesure. Parmi ces objets, nous citerons **ambitus**, **breathing-sign**, **clef**, **custos**, **staff-bar**, **left-edge**, **key-cancellation**, **key-signature**, et **time-signature**.

Ces indications et numéros de mesure sont par défaut centrés horizontalement au-dessus de l'objet :

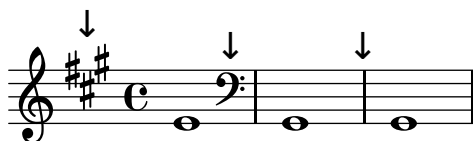
```
% The rehearsal mark will be centered above the Clef
\override Score.RehearsalMark #'break-align-symbols = #'(clef)
\key a \major
\clef treble
\mark ""
e1
% The rehearsal mark will be centered above the Time Signature
\override Score.RehearsalMark #'break-align-symbols = #'(time-signature)
\key a \major
\clef treble
\time 3/4
\mark ""
e2.
% The rehearsal mark will be centered above the Breath Mark
\override Score.RehearsalMark #'break-align-symbols = #'(breathing-sign)
\key a \major
\clef treble
\time 4/4
e1
\breathe
\mark ""
```



Les différents objets sur lesquels l'alignement pourrait intervenir seront regroupés dans une liste. Si l'un des objets est invisible à l'endroit voulu, en raison d'un réglage de **break-visibility** ou bien par forçage de la visibilité des armures et clefs, le repère ou le numéro de mesure viendra s'aligner sur le premier élément de cette liste qui soit visible. Dans le cas où aucun objet de la liste n'est visible, l'alignement se fera sur la barre de mesure ou, dans le cas où la barre de mesure est invisible, à l'endroit même où la barre prendrait place.

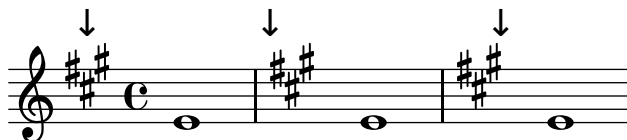
```
% The rehearsal mark will be centered above the Key Signature
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark ""
e1
% The rehearsal mark will be centered above the Clef
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature clef)
\key a \major
\clef bass
\mark ""
```

```
gis,,1
% The rehearsal mark will be centered above the Bar Line
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.explicitClefVisibility = #all-invisible
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark ""
e''1
```



L'alignement d'un repère sur un objet de notation peut se modifier, comme l'illustre l'exemple suivant. Toutefois, si la partition comporte plusieurs portées, ce réglage devra apparaître dans chacune des portées.

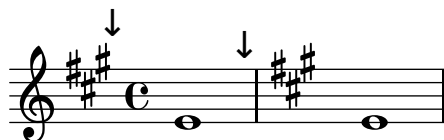
```
% The RehearsalMark will be centered above the Key Signature
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature)
\key a \major
\clef treble
\time 4/4
\mark ""
e1
% The RehearsalMark will be aligned with the left edge of the Key Signature
\once \override Score.KeySignature #'break-align-anchor-alignment = #LEFT
\mark ""
\key a \major
e1
% The RehearsalMark will be aligned with the right edge of the Key Signature
\once \override Score.KeySignature #'break-align-anchor-alignment = #RIGHT
\key a \major
\mark ""
e1
```



Le bord gauche d'un repère peut se décaler arbitrairement sur la gauche ou la droite. La valeur est exprimée en espaces de portée.

```
% The RehearsalMark will be aligned with the left edge of the Key Signature
% and then shifted right by 3.5 staff-spaces
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature)
\once \override Score.KeySignature #'break-align-anchor = #3.5
\key a \major
\mark ""
e1
% The RehearsalMark will be aligned with the left edge of the Key Signature
% and then shifted left by 2 staff-spaces
```

```
\once \override Score.KeySignature #'break-align-anchor = #-2
\key a \major
\mark ""
e1
```



5.5.2 Regroupement vertical d'objets graphiques

Les objets `VerticalAlignment` et `VerticalAxisGroup` travaillent de concert. Comme leur nom anglais l'indiquent, `VerticalAxisGroup` regroupe différents objets tels que les portées (`Staff`), les paroles (`Lyrics`) et ainsi de suite ; puis `VerticalAlignment` synchronise verticalement ces différents groupes. En général, il n'y a qu'un seul `VerticalAlignment` pour l'ensemble de la partition, mais chaque contexte `Staff`, `Lyrics`, etc. possède son propre `VerticalAxisGroup`.

5.5.3 Modification des stencils

Tout objet de rendu dispose d'une propriété `stencil` attachée à la `grob-interface`. En règle générale, cette propriété référence par défaut une fonction spécifique à l'objet et taillée sur mesure pour fournir le symbole qui va le représenter dans l'output. Par exemple, le réglage standard de la propriété `stencil` de l'objet `MultiMeasureRest` est `ly:multi-measure-rest::print`.

Le symbole standard d'un objet quel qu'il soit peut être remplacé à partir du moment où la propriété `stencil` référence une procédure différente et écrite à cet effet. Ceci requiert une bonne maîtrise du fonctionnement interne de LilyPond, mais est grandement facilité dans bien des cas et permet d'obtenir le résultat escompté.

En effet, rien ne nous interdit, à partir de la propriété `stencil`, d'appeler la procédure qui génère du texte, `ly:text-interface::print` en l'occurrence, et d'adjoindre à l'objet une propriété `text` qui contiendra, sous forme de *markup*, le symbole à dessiner. Grâce à l'extrême flexibilité des *markups*, vous pourrez parvenir à bien des choses – voir à ce sujet [\[Éléments graphiques dans du texte formaté\]](#), page 228.

C'est la technique employée ici, où l'une des têtes de note est remplacée par une croix inscrite dans un cercle :

```
Xin0 = {
  \once \override NoteHead #'stencil = #ly:text-interface::print
  \once \override NoteHead #'text = \markup {
    \combine
      \halign #-0.7 \draw-circle #0.85 #0.2 ##f
      \musicglyph #"noteheads.s2cross"
  }
}
\relative c'' {
  a a \Xin0 a a
}
```



Tous les glyphes de la fonte Feta sont accessibles à l'aide de la commande de *markup* `\musicglyph` – voir [Section A.8 \[La fonte Feta\]](#), page 620.

Voir aussi

Manuel de notation : [\[Text markup commands\]](#), page [\[undefined\]](#), [\[Éléments graphiques dans du texte formaté\]](#), page 228, Section A.8 [\[La fonte Feta\]](#), page 620, Section 1.8.2 [\[Mise en forme du texte\]](#), page 221.

5.5.4 Modification de l'allure des éléments

Modification des liaisons

Les liaisons, qu'elles soient de prolongation, d'articulation ou de phrasé, sont dessinées sous la forme de courbes de Bézier de degré trois. Lorsque l'aspect de la liaison automatiquement calculé n'est pas satisfaisant, il peut être modifié manuellement, en fournissant explicitement les quatre points de contrôle qui permettront de définir cette courbe.

Quatre points définissent une courbe de Bézier cubique. Les premier et quatrième points sont les points de départ et d'arrivée de la courbe ; les deux autres points de contrôle – P1 et P2 – en détermineront l'allure. La courbe se trace en partant du point P0, en se dirigeant vers P1 et en arrivant au point P3 selon la direction P2-P3. La courbe est à l'intérieur de l'enveloppe convexe des points de contrôle.

Voici par exemple une liaison de prolongation dont l'allure n'est pas des plus heureuses, même en optant pour un `\tieDown`.

```
<<
{ e1~ e }
\\
{ r4 <g c,> <g c,> <g c,> }
>>
```



L'un des moyens d'améliorer cette liaison consiste à modifier manuellement ses quatre points de contrôle.

Les coordonnées des points de contrôle sont données en unités d'espace de portée. L'abscisse est relative au point de référence de la note de départ de la liaison ; l'ordonnée est relative à la ligne médiane de la portée. Les différentes coordonnées sont entrées sous la forme d'une liste de quatre paires de nombres décimaux (ou nombres réels). L'une des manières de procéder consiste à tout d'abord estimer les coordonnées des deux extrémités puis, par tâtonnement, déterminer les deux points intermédiaires.

Remarque intéressante au sujet des courbes : une courbe sera symétrique si ses points de contrôle sont symétriques. L'une des propriétés avantageuses des courbes de Bézier est que leur transformation – déplacement, rotation ou échelonnement – est réalisable en simplement corrigeant de manière identique ses points de contrôle.

Pour l'exemple qui nous occupe, l'adaptation suivante nous permet d'obtenir un résultat plutôt satisfaisant. Notez bien l'endroit où cette adaptation est stipulée : juste avant la note de départ de la liaison.

```
<<
{
\once \override Tie
  #'control-points = #'((1 . -1) (3 . 0.6) (12.5 . 0.6) (14.5 . -1))
  e1 ~ e
}
```

```

}
\\
{ r4 <g c,> <g c,> <g c,> }
>>

```



Problèmes connus et avertissements

Lorsque plusieurs liaisons, quelle qu'en soit la nature, commencent au même moment, jouer sur la propriété `control-points` est impossible, et la commande `\tweak` inefficace. Vous pouvez néanmoins influencer sur la propriété `tie-configuration` de l'objet `TieColumn` pour déterminer la ligne de départ et l'orientation.

Voir aussi

Référence des propriétés internes : [Section “TieColumn” dans Référence des propriétés internes](#).

5.5.5 Conteneurs requalifiants

Les conteneurs requalifiants permettent de faciliter le calcul des espacements en cas de modification du *Y-axis* – plus particulièrement les composantes `Y-offset` et `Y-extent` – à l'aide d'une fonction `schema` en lieu et place de valeurs.

L'envergure verticale (`Y-extent`) de certains objets dépend de la propriété `stencil` ; jouer sur leur `stencil` requiert alors une intervention supplémentaire au niveau du `Y-extent` à l'aide d'un conteneur transitoire. Lorsqu'une fonction affecte un `Y-offset` ou un `Y-extent`, cela déclenche la détermination des sauts de ligne de manière anticipée dans la séquence des traitements. Il en résulte que cette opération n'est en fait pas exécutée ; elle renvoie habituellement 0 ou '(0 . 0), ce qui peut engendrer des collisions. Une fonction « pure » évitera d'avorter la construction des propriétés ou objets, qui de ce fait verront leurs arguments liés à la verticalité (`Y-axis`) correctement évalués.

Il existe actuellement une trentaine de fonctions que l'on peut qualifier de « pures ». Le recours à un conteneur transitoire permet de requalifier une fonction de telle sorte qu'elle soit reconnue comme « pure » et soit donc évaluée **avant** détermination des sauts de ligne – l'espacement horizontal sera de fait ajusté en temps et en heure. La fonction « impure » sera ensuite évaluée **après** le positionnement des sauts de ligne.

Note : Il n'est pas toujours facile d'avoir l'assurance qu'une fonction soit qualifiée de « pure » ; aussi nous vous recommandons d'éviter d'utiliser les objets `Beam` or `VerticalAlignment` lorsque vous désirez en créer une.

Un conteneur requalifiant se construit selon la syntaxe

```
(ly:make-unpure-pure-container f0 f1)
```

où `f0` est une fonction prenant n arguments ($n \geq 1$), le premier devant être l'objet en question ; il s'agit de la fonction dont le résultat sera réutilisé. `f1` est la fonction qui sera qualifiée de « pure ». Elle prend $n+2$ arguments, le premier devant être lui aussi l'objet en question, et les second et troisième étant respectivement les « point de départ » (*start*) et « point d'arrivée » (*end*).

start et *end* sont dans tous les cas des valeurs fictives qui trouveront leur utilité dans le cas d'objets de type **Spanner**, tels les soufflets (**Hairpin**) ou barres de ligature (**Beam**), en retournant les différentes estimations de hauteur basées sur leurs début et fin d'extension.

Viennent ensuite les autres arguments de la fonction initiale **f0** – autrement dit aucun si *n_* = -1.

Les résultats de la deuxième fonction (**f1**) permettent une approximation des valeurs qui seront ensuite utilisées par la fonction initiale aux fins d'ajustement lors des phases ultérieures d'espacement.

```
#(define (square-line-circle-space grob)
(let* ((pitch (ly:event-property (ly:grob-property grob 'cause) 'pitch))
      (notename (ly:pitch-notename pitch)))
  (if (= 0 (modulo notename 2))
      (make-circle-stencil 0.5 0.0 #t)
      (make-filled-box-stencil '(0 . 1.0)
                               '(-0.5 . 0.5))))))

squareLineCircleSpace = {
  \override NoteHead #'stencil = #square-line-circle-space
}

smartSquareLineCircleSpace = {
  \squareLineCircleSpace
  \override NoteHead #'Y-extent =
    #(ly:make-unpure-pure-container
      ly:grob::stencil-height
      (lambda (grob start end) (ly:grob::stencil-height grob)))
}

\new Voice \with { \remove "Stem_engraver" }
\relative c'' {
  \squareLineCircleSpace
  cis4 ces cisis c
  \smartSquareLineCircleSpace
  cis4 ces cisis c
}
```



La première mesure de l'exemple ci-dessus ne fait pas appel à un conteneur requalifiant ; le moteur d'espacement n'a donc aucune connaissance de la largeur des têtes de note et ne peut empêcher qu'elles chevauchent les altérations. Dans la deuxième mesure, par contre, le recours à un conteneur requalifiant informe le moteur d'espacement de la largeur des têtes de note ; les collisions sont alors évitées du fait de l'espace réservé à chacune des têtes.

Lorsqu'il s'agit de calculs simples, les fonctions, tant pour la partie « pure » que pour la partie « impure », peuvent être identiques au détail près du nombre d'arguments utilisés ou du domaine d'intervention.

Note : Le fait de qualifier une fonction de « pure » alors qu'elle ne l'est pas peut générer des résultats imprévisibles.

5.6 Utilisation de fonctions musicales

Une adaptation ou un affinage qui devient récurrent parce que doit s'appliquer à différentes expressions musicales peut faire l'objet d'une *fonction musicale*. Nous ne traiterons ici que des fonctions de *substitution*, dont le but est de substituer une variable en un bout de code LilyPond. D'autres fonctions, plus complexes, sont abordées au chapitre [Section “Fonctions musicales” dans *Extension de LilyPond*](#).

5.6.1 Syntaxe d'une fonction de substitution

La rédaction d'une fonction chargée de substituer du code LilyPond à une variable est chose relativement aisée. Une telle fonction est de la forme

```
fonction =
#(define-music-function
  (parser location arg1 arg2 ...)
  (type1? type2? ...)
  #{
    ...musique...
  #})
```

où

argN *n*ème argument.

typeN? un *type de prédicat* Scheme pour lequel *argN* doit renvoyer *#t*.

...musique... du code LilyPond tout ce qu'il y a de plus ordinaire, avec des *\$* (là où seule une construction LilyPond est autorisée) et des *#* (lorsqu'il s'agit d'une valeur en Scheme, d'un argument de fonction musicale ou de musique faisant partie d'une liste) pour référencer les arguments (par ex. *'#arg1'*).

Les arguments *parser* et *location* sont obligatoires ; ils sont utilisés dans certaines situations évoluées, comme indiqué dans le manuel d'extension des fonctionnalités à au chapitre [Section “Fonctions musicales” dans *Extension de LilyPond*](#). Assurez-vous bien de ne pas les omettre dans vos fonctions de substitution.

La liste des types de prédicat est elle aussi obligatoire. Voici quelques uns des types de prédicat les plus utilisés dans les fonctions musicales :

```
boolean?
cheap-list? (au lieu de « list? »
, pour accélérer le traitement)
ly:duration?
ly:music?
ly:pitch?
markup?
number?
pair?
string?
symbol?
```

Une liste plus fournie est disponible à l'annexe [Section A.20 \[Types de prédicats prédéfinis\], page 729](#). Vous pouvez par ailleurs définir vos propres types de prédicat.

Voir aussi

Manuel de notation : [Section A.20 \[Types de prédicats prédéfinis\], page 729](#).

Manuel d'extension : [Section “Fonctions musicales”](#) dans *Extension de LilyPond*.

Fichiers d'initialisation : ‘lily/music-scheme.cc’, ‘scm/c++.scm’, ‘scm/lily.scm’.

5.6.2 Exemples de fonction de substitution

La présente rubrique regroupe quelques exemples de fonction substitutive. Le propos est ici d'illustrer les possibilités qu'offrent les fonctions de substitution simple.

Dans ce premier exemple, nous définissons une fonction dans le but de simplifier le réglage du décalage d'une annotation (un `TextScript`).

```
padText =
#(define-music-function
  (parser location padding)
  (number?)
  #{
    \once \override TextScript #'padding = #padding
  #})

\relative c''' {
  c4~"piu mosso" b a b
  \padText #1.8
  c4~"piu mosso" d e f
  \padText #2.6
  c4~"piu mosso" fis a g
}
```



Nous pouvons utiliser autre chose que des nombres au sein d'une fonction, y compris une expression musicale :

```
custosNote =
#(define-music-function
  (parser location note)
  (ly:music?)
  #{
    \tweak NoteHead #'stencil #ly:text-interface::print
    \tweak NoteHead #'text
      \markup \musicglyph #"custodes.mensural.u0"
    \tweak Stem #'stencil ##f
    #note
  #})

\relative c' { c4 d e f \custosNote g }
```



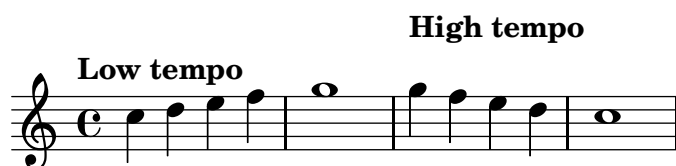
Une fonction de substitution peut traiter plusieurs arguments :

```

tempoPadded =
#(define-music-function
  (parser location padding tempotext)
  (number? markup?)
  #{
    \once \override Score.MetronomeMark #'padding = #padding
    \tempo \markup { \bold #tempotext }
  #})

\relative c'' {
  \tempo \markup { "Low tempo" }
  c4 d e f g1
  \tempoPadded #4.0 "High tempo"
  g4 f e d c1
}

```



Annexe A Tables du manuel de notation

A.1 Table des noms d'accord

La liste suivante répertorie les noms des accords selon les deux types les plus répandus, ainsi que ce qu'ils représentent.

Ignatzek (default)

Alternative

	C	Cm	C+	C ^o
	C	C ^{b3}	C ^{#5}	C ^{b3 b5}
Def	C ⁷	Cm ⁷	C ^Δ	C ^{o7}
Alt ⁵	C ⁷	C ^{7 b3}	C ^{#7}	C ^{b3 b5 b7}
Def	C ^{7 #5}	Cm ^Δ	C ^{Δ #5}	C [∅]
Alt ¹⁰	C ^{7 #5}	C ^{b3 #7}	C ^{#5 #7}	C ^{7 b3 b5}
Def	C ⁶	Cm ⁶	C ⁹	Cm ⁹
Alt ¹⁴	C ⁶	C ^{b3 6}	C ⁹	C ^{9 b3}
Def	Cm ¹³	Cm ¹¹	Cm ^{7 b5 9}	C ^{7 b9}
Alt ⁸	C ^{13 b3}	C ^{11 b3}	C ^{9 b3 b5}	C ^{7 b9}
Def	C ^{7 #9}	C ¹¹	C ^{7 #11}	C ¹³
Alt ²	C ^{7 #9}	C ¹¹	C ^{9 #11}	C ¹³
Def	C ^{7 #11 b13}	C ^{7 #5 #9}	C ^{7 #9 #11}	C ^{7 b13}
Alt ²⁶	C ^{9 #11 b13}	C ^{7 #5 #9}	C ^{7 #9 #11}	C ^{11 b13}

Def	$C^{7\flat 9\flat 13}$	$C^{7\sharp 11}$	$C^{\triangle 9}$	$C^{7\flat 13}$
Alt ₃₀	$C^{11\flat 9\flat 13}$	$C^9\sharp 11$	$C^9\sharp 7$	$C^{11\flat 13}$
				
Def	$C^{7\flat 9\flat 13}$	$C^{7\flat 9\flat 13}$	$C^{\triangle 9}$	$C^{\triangle 13}$
Alt ₃₄	$C^{11\flat 9\flat 13}$	$C^{13\flat 9}$	$C^9\sharp 7$	$C^{13\sharp 7}$
				
Def	$C^{\triangle \sharp 11}$	$C^{7\flat 9\flat 13}$	C^{sus4}	C^{7sus4}
Alt ₃₈	$C^9\sharp 7\sharp 11$	$C^{13\flat 9}$	$C^{add4\ 5}$	$C^{add4\ 5\ 7}$
				
Def	C^9sus4	C^9	C^{m11}	C^{lyd}
Alt ₄₂	$C^{add4\ 5\ 7\ 9}$	C^{add9}	$C^{\flat 3\ add11}$	$C^{\sharp 7\ add\sharp 11}$
				

A.2 Modificateurs d'accord usuels

Le tableau suivant indique les différents modificateurs utilisables en mode accords (`\chordmode`) qui permettent d'obtenir les structures habituelles d'un certain nombre d'accords.

Type d'accord	Intervalles	Modificateur(s)	Exemple
---------------	-------------	-----------------	---------

Majeur

Tierce majeure et quinte
juste

5 ou rien



Mineur

Tierce mineure et quinte
juste

m ou m5



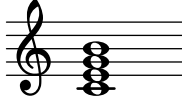
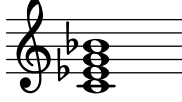
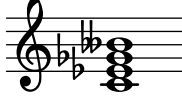
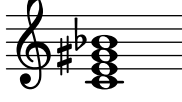


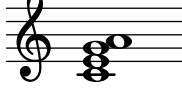



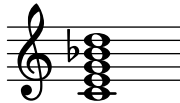
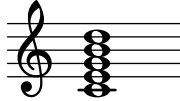
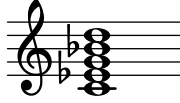
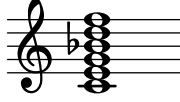
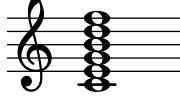
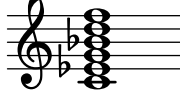




Augmenté


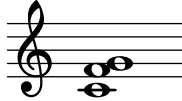
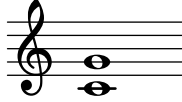
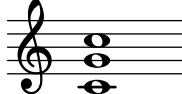
Tierce majeure et quinte
augmentée

aug



Diminué	Tierce mineure et quinte diminuée	dim	
Septième de dominante	Triton majeur et septième mineure	7	
Septième majeure	Triton majeur et septième majeure	maj7 ou maj	
Septième mineure	Triton mineur et septième mineure	m7	
Septième diminuée	Triton diminué et septième diminuée	dim7	
Septième augmentée	Triton augmenté et septième mineure	aug7	
Septième semi-diminuée	Triton diminué et septième mineure	m7.5-	
Accord mineur avec septième majeure	Triton mineur et septième majeure	m7+	
Sixte majeure	Triton majeur et sixte	6	
Sixte mineure	Triton mineur et sixte	m6	

Neuvième de dominante	Septième de dominante et neuvième majeure	9	
Neuvième majeure	Septième majeure et neuvième majeure	maj9	
Neuvième mineure	Septième mineure et neuvième majeure	m9	
Onzième de dominante	Neuvième de dominante et onzième juste	11	
Onzième majeure	Neuvième majeure et onzième juste	maj11	
Onzième mineure	Neuvième mineure et onzième juste	m11	
Treizième de dominante	Neuvième de dominante et treizième majeure	13	
Treizième de dominante	Onzième de dominante et treizième majeure	13.11	
Treizième majeure	Onzième majeure et treizième majeure	maj13.11	
Treizième mineure	Onzième mineure et treizième majeure	m13.11	

Seconde suspendue	Seconde majeure et quinte juste	sus2	
Quarte suspendue	Quarte juste et quinte juste	sus4	
Power chord (deux voix)	Quinte juste	1.5	
Power chord (trois voix)	Quinte juste et octave	1.5.8	

A.3 Accordages prédéfinis

La liste suivante répertorie les différents accordages dont LilyPond dispose.

Guitar tunings

8 guitar-tuning guitar-seven-string-tuning guitar-drop-d-tuning

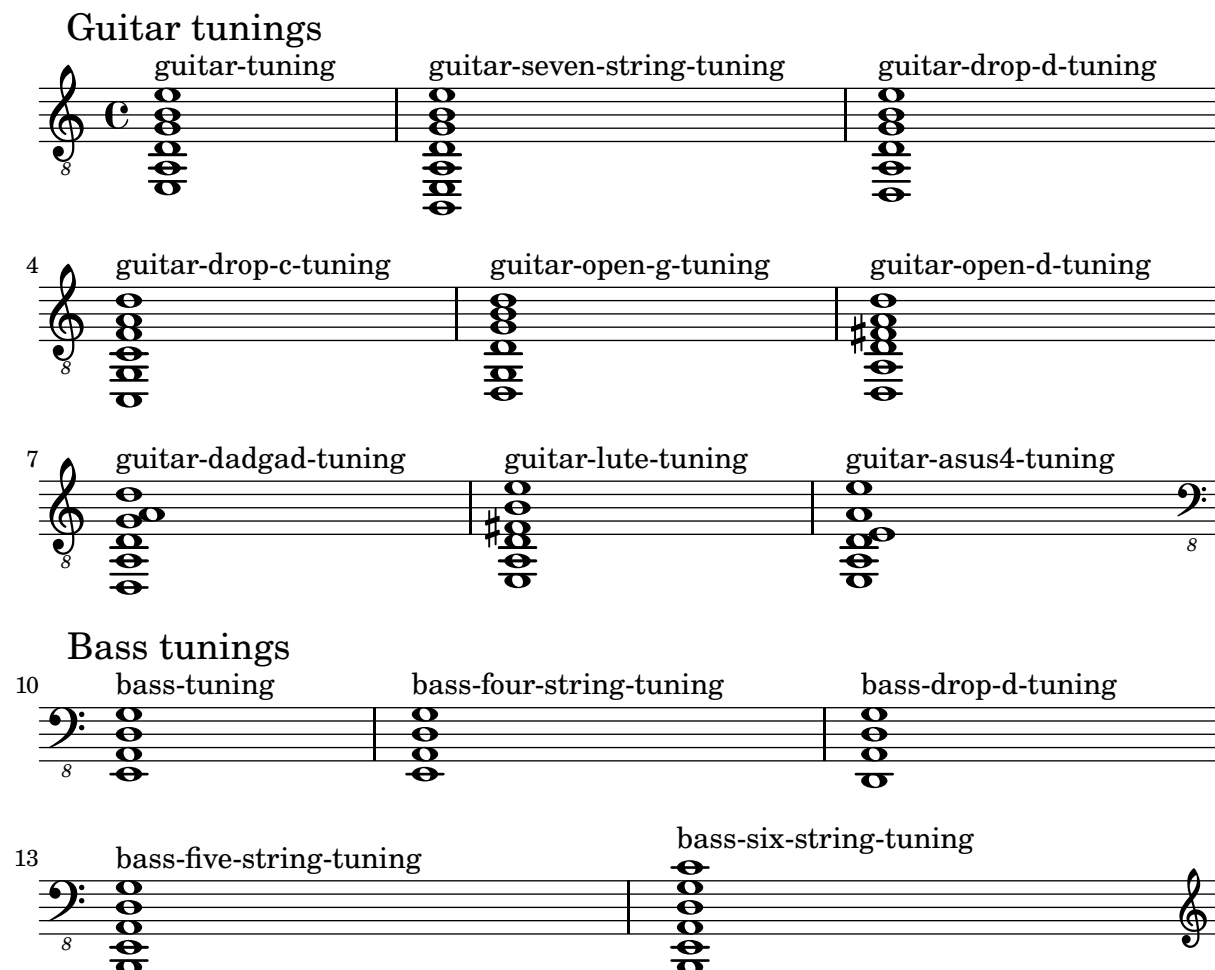
4 guitar-drop-c-tuning guitar-open-g-tuning guitar-open-d-tuning

7 guitar-dadgad-tuning guitar-lute-tuning guitar-asus4-tuning

Bass tunings

10 bass-tuning bass-four-string-tuning bass-drop-d-tuning

13 bass-five-string-tuning bass-six-string-tuning



Mandolin tunings

15 mandolin-tuning

Banjo tunings

16 banjo-open-g-tuning

banjo-c-tuning

18 banjo-modal-tuning

banjo-open-d-tuning

banjo-open-dm-tuning

Ukulele tunings

21 ukulele-tuning

ukulele-d-tuning

23 tenor-ukulele-tuning

baritone-ukulele-tuning

Orchestral string tunings

25 violin-tuning

viola-tuning

cello-tuning

double-bass-tuning

A.4 Diagrammes d'accord prédéfinis

La liste suivante répertorie les tablatures prédéfinies pour la guitare.

Diagrammes d'accord prédéfinis pour la guitare (guitare standard 6 cordes).

10

C Cm C+ C^o C^{o7} C⁷ C^Δ Cm⁷ C⁹

19

D^b D^bm D^b+ D^{b o} D^{b o7} D^{b 7} D^{b Δ} D^{b m 7} D^{b 9}

Diagrammes d'accord prédéfinis pour la guitare (guitare standard 6 cordes).

Diagrammes d'accord prédéfinis pour la guitare (guitare standard 6 cordes).

28	D	Dm	D+	D ⁰	D ⁰⁷	D ⁷	D ^Δ	Dm ⁷	D ⁹
37	D [#]	D [#] m	D [#] +	D ^{#0}	D ^{#07}	D ^{#7}	D ^{#Δ}	D [#] m ⁷	D ^{#9}
46	E _b	E _b m	E _b +	E _b ⁰	E _b ⁰⁷	E _b ⁷	E _b ^Δ	E _b m ⁷	E _b ⁹
55	E	Em	E+	E ⁰	E ⁰⁷	E ⁷	E ^Δ	Em ⁷	E ⁹
64	F	Fm	F+	F ⁰	F ⁰⁷	F ⁷	F ^Δ	Fm ⁷	F ⁹
73	F [#]	F [#] m	F [#] +	F ^{#0}	F ^{#07}	F ^{#7}	F ^{#Δ}	F [#] m ⁷	F ^{#9}
82	G _b	G _b m	G _b +	G _b ⁰	G _b ⁰⁷	G _b ⁷	G _b ^Δ	G _b m ⁷	G _b ⁹
91	G	Gm	G+	G ⁰	G ⁰⁷	G ⁷	G ^Δ	Gm ⁷	G ⁹
100	G [#]	G [#] m	G [#] +	G ^{#0}	G ^{#07}	G ^{#7}	G ^{#Δ}	G [#] m ⁷	G ^{#9}
109	A _b	A _b m	A _b +	A _b ⁰	A _b ⁰⁷	A _b ⁷	A _b ^Δ	A _b m ⁷	A _b ⁹

118 **A** **Am** **A+** **A^o** **A^{o7}** **A⁷** **A[△]** **Am⁷** **A⁹**

127 **A[#]** **A[#]m** **A[#]+** **A^{#o}** **A^{#o7}** **A^{#7}** **A^{#△}** **A^{#m7}** **A^{#9}**

136 **B^b** **B^bm** **B^b+** **B^{b0}** **B^{b07}** **B^{b7}** **B^{b△}** **B^{b7}** **B^{b9}**

145 **B** **Bm** **B+** **B^o** **B^{o7}** **B⁷** **B[△]** **Bm⁷** **B⁹**

La liste suivante répertorie les tablatures prédéfinies pour le ukulele.

C **Cm** **C+** **C^o** **C⁷** **C[△]** **Cm⁷** **C⁶** **C^{sus2}** **C^{sus4}** **C⁹**

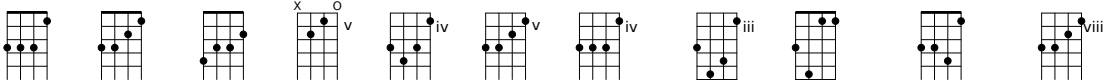
12 **C[#]** **C[#]m** **C[#]+** **C^{#o}** **C^{#7}** **C^{#△}** **C^{#m7}** **C^{#6}** **C^{#sus2}** **C^{#sus4}** **C^{#9}**

23 **D^b** **D^bm** **D^b+** **D^{b0}** **D^{b7}** **D^{b△}** **D^{b7}** **D^{b6}** **D^bsus2** **D^bsus4** **D^{b9}**

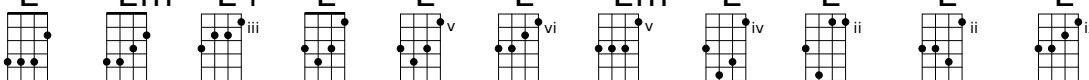
34 **D** **Dm** **D+** **D^o** **D⁷** **D[△]** **Dm⁷** **D⁶** **D^{sus2}** **D^{sus4}** **D⁹**

45 **D[#]** **D[#]m** **D[#]+** **D^{#o}** **D^{#7}** **D^{#△}** **D^{#m7}** **D^{#6}** **D^{#sus2}** **D^{#sus4}** **D^{#9}**

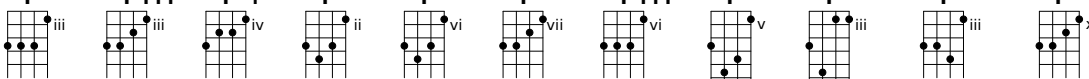
56 E_b $E_b m$ $E_b +$ E_b^o E_b^7 E_b^{\triangle} $E_b m^7$ E_b^6 E_b^{sus2} E_b^{sus4} E_b^9



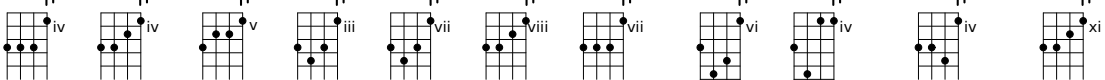
67 E $E m$ $E +$ E^o E^7 E^{\triangle} $E m^7$ E^6 E^{sus2} E^{sus4} E^9



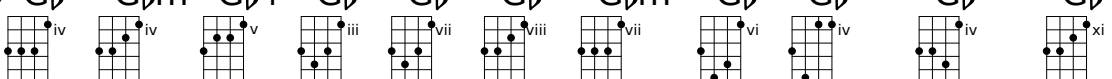
78 F $F m$ $F +$ F^o F^7 F^{\triangle} $F m^7$ F^6 F^{sus2} F^{sus4} F^9



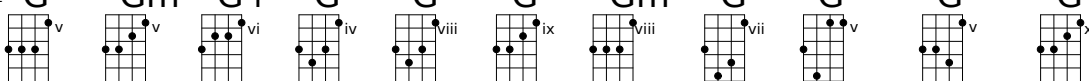
89 $F^{\#}$ $F^{\#} m$ $F^{\#} +$ $F^{\# o}$ $F^{\# 7}$ $F^{\# \triangle}$ $F^{\#} m^7$ $F^{\# 6}$ $F^{\# sus2}$ $F^{\# sus4}$ $F^{\# 9}$



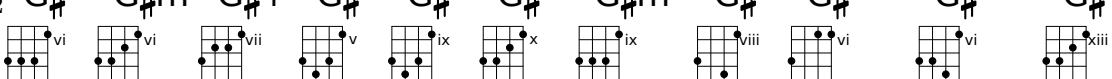
100 G_b $G_b m$ $G_b +$ G_b^o G_b^7 G_b^{\triangle} $G_b m^7$ G_b^6 G_b^{sus2} G_b^{sus4} G_b^9



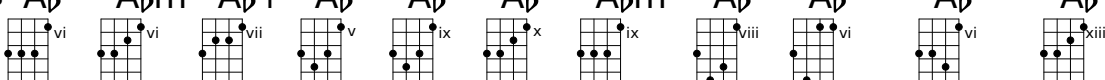
111 G $G m$ $G +$ G^o G^7 G^{\triangle} $G m^7$ G^6 G^{sus2} G^{sus4} G^9



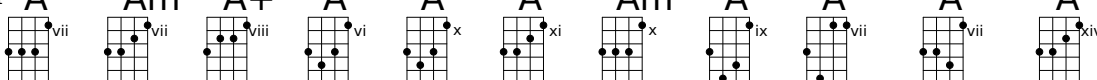
122 $G^{\#}$ $G^{\#} m$ $G^{\#} +$ $G^{\# o}$ $G^{\# 7}$ $G^{\# \triangle}$ $G^{\#} m^7$ $G^{\# 6}$ $G^{\# sus2}$ $G^{\# sus4}$ $G^{\# 9}$



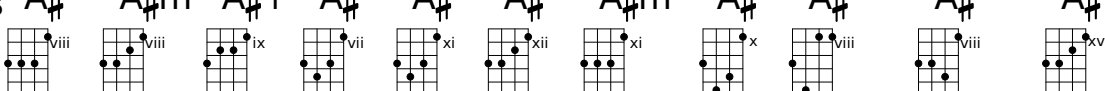
133 A_b $A_b m$ $A_b +$ A_b^o A_b^7 A_b^{\triangle} $A_b m^7$ A_b^6 A_b^{sus2} A_b^{sus4} A_b^9



144 A $A m$ $A +$ A^o A^7 A^{\triangle} $A m^7$ A^6 A^{sus2} A^{sus4} A^9



155 $A^{\#}$ $A^{\#} m$ $A^{\#} +$ $A^{\# o}$ $A^{\# 7}$ $A^{\# \triangle}$ $A^{\#} m^7$ $A^{\# 6}$ $A^{\# sus2}$ $A^{\# sus4}$ $A^{\# 9}$



166 $B\flat$ $B\flat m$ $B\flat +$ $B\flat^o$ $B\flat^7$ $B\flat^\Delta$ $B\flat m^7$ $B\flat^6$ $B\flat^{sus2}$ $B\flat^{sus4}$ $B\flat^9$

177 B Bm $B+$ B^o B^7 B^Δ Bm^7 B^6 B^{sus2} B^{sus4} B^9

La liste suivante répertorie les tablatures prédéfinies pour la mandoline.

C Cm $C+$ C^{o7} C^7 C^Δ Cm^7 C^\emptyset C^6 C^{sus2} C^{sus4} C^9

13 C^\sharp $C^\sharp m$ $C^\sharp +$ C^\sharp^{o7} C^\sharp^7 C^\sharp^Δ $C^\sharp m^7$ C^\sharp^\emptyset C^\sharp^6 C^\sharp^{sus2} C^\sharp^{sus4} C^\sharp^9

25 $D\flat$ $D\flat m$ $D\flat +$ $D\flat^{o7}$ $D\flat^7$ $D\flat^\Delta$ $D\flat m^7$ $D\flat^\emptyset$ $D\flat^6$ $D\flat^{sus2}$ $D\flat^{sus4}$ $D\flat^9$

37 D Dm $D+$ D^{o7} D^7 D^Δ Dm^7 D^\emptyset D^6 D^{sus2} D^{sus4} D^9

49 D^\sharp $D^\sharp m$ $D^\sharp +$ D^\sharp^{o7} D^\sharp^7 D^\sharp^Δ $D^\sharp m^7$ D^\sharp^\emptyset D^\sharp^6 D^\sharp^{sus2} D^\sharp^{sus4} D^\sharp^9

61 $E\flat$ $E\flat m$ $E\flat +$ $E\flat^{o7}$ $E\flat^7$ $E\flat^\Delta$ $E\flat m^7$ $E\flat^\emptyset$ $E\flat^6$ $E\flat^{sus2}$ $E\flat^{sus4}$ $E\flat^9$

73 E Em $E+$ E^{o7} E^7 E^Δ Em^7 E^\emptyset E^6 E^{sus2} E^{sus4} E^9

85 **F** **Fm** **F+** **F^{o7}** **F⁷** **F[△]** **Fm⁷** **F[∅]** **F⁶** **F^{sus2}** **F^{sus4}** **F⁹**

97 **F[#]** **F[#]m** **F[#]+** **F^{#o7}** **F^{#7}** **F^{#△}** **F^{#m7}** **F^{#∅}** **F^{#6}** **F^{#sus2}** **F^{#sus4}** **F^{#9}**

109 **G^b** **G^bm** **G^b+** **G^{b o7}** **G^{b7}** **G^{b△}** **G^{b m7}** **G^{b∅}** **G^{b6}** **G^{b sus2}** **G^{b sus4}** **G^{b9}**

121 **G** **Gm** **G+** **G^{o7}** **G⁷** **G[△]** **Gm⁷** **G[∅]** **G⁶** **G^{sus2}** **G^{sus4}** **G⁹**

133 **G[#]** **G[#]m** **G[#]+** **G^{# o7}** **G^{#7}** **G^{#△}** **G^{# m7}** **G^{#∅}** **G^{#6}** **G^{# sus2}** **G^{# sus4}** **G^{#9}**

145 **A^b** **A^bm** **A^b+** **A^{b o7}** **A^{b7}** **A^{b△}** **A^{b m7}** **A^{b∅}** **A^{b6}** **A^{b sus2}** **A^{b sus4}** **A^{b9}**

157 **A** **Am** **A+** **A^{o7}** **A⁷** **A[△]** **Am⁷** **A[∅]** **A⁶** **A^{sus2}** **A^{sus4}** **A⁹**

169 **A[#]** **A[#]m** **A[#]+** **A^{# o7}** **A^{#7}** **A^{#△}** **A^{# m7}** **A^{#∅}** **A^{#6}** **A^{# sus2}** **A^{# sus4}** **A^{#9}**

181 **B^b** **B^bm** **B^b+** **B^{b o7}** **B^{b7}** **B^{b△}** **B^{b m7}** **B^{b∅}** **B^{b6}** **B^{b sus2}** **B^{b sus4}** **B^{b9}**

193 **B** **Bm** **B+** **B^{o7}** **B⁷** **B[△]** **Bm⁷** **B[∅]** **B⁶** **B^{sus2}** **B^{sus4}** **B⁹**

A.5 Formats de papier prédéfinis

Les formats de page sont définis dans le fichier ‘scm/paper.scm’.

La série A « ISO 216 »

"a10"	(26 x 37 mm)
"a9"	(37 x 52 mm)
"a8"	(52 x 74 mm)
"a7"	(74 x 105 mm)
"a6"	(105 x 148 mm)
"a5"	(148 x 210 mm)
"a4"	(210 x 297 mm)
"a3"	(297 x 420 mm)
"a2"	(420 x 594 mm)
"a1"	(594 x 841 mm)
"a0"	(841 x 1189 mm)

La série B « ISO 216 »

"b10"	(31 x 44 mm)
"b9"	(44 x 62 mm)
"b8"	(62 x 88 mm)
"b7"	(88 x 125 mm)
"b6"	(125 x 176 mm)
"b5"	(176 x 250 mm)
"b4"	(250 x 353 mm)
"b3"	(353 x 500 mm)
"b2"	(500 x 707 mm)
"b1"	(707 x 1000 mm)
"b0"	(1000 x 1414 mm)

Deux tailles étendues, définies par la « DIN 476 »

"4a0"	(1682 x 2378 mm)
"2a0"	(1189 x 1682 mm)

La série C standard « ISO 269 »

"c10"	(28 x 40 mm)
"c9"	(40 x 57 mm)
"c8"	(57 x 81 mm)
"c7"	(81 x 114 mm)
"c6"	(114 x 162 mm)
"c5"	(162 x 229 mm)
"c4"	(229 x 324 mm)
"c3"	(324 x 458 mm)
"c2"	(458 x 648 mm)
"c1"	(648 x 917 mm)
"c0"	(917 x 1297 mm)

Formats nord américains

"junior-legal"	(8.0 x 5.0 in)
"legal"	(8.5 x 14.0 in)
"ledger"	(17.0 x 11.0 in)
"letter"	(8.5 x 11.0 in)
"tabloid"	(11.0 x 17.0 in)
"11x17"	(11.0 x 17.0 in)
"17x11"	(17.0 x 11.0 in)

Government-letter, défini par le *IEEE Printer Working Group*, à l'usage des enfants

"government-letter"	(8 x 10.5 in)
"government-legal"	(8.5 x 13.0 in)
"philippine-legal"	(8.5 x 13.0 in)

Formats ANSI

"ansi a"	(8.5 x 11.0 in)
"ansi b"	(17.0 x 11.0 in)
"ansi c"	(17.0 x 22.0 in)
"ansi d"	(22.0 x 34.0 in)
"ansi e"	(34.0 x 44.0 in)
"engineering f"	(28.0 x 40.0 in)

Formats nord américains pour l'architecture

"arch a"	(9.0 x 12.0 in)
"arch b"	(12.0 x 18.0 in)
"arch c"	(18.0 x 24.0 in)
"arch d"	(24.0 x 36.0 in)
"arch e"	(36.0 x 48.0 in)
"arch e1"	(30.0 x 42.0 in)

Formats anciens, toujours en vigueur dans le Royaume Uni

"statement"	(5.5 x 8.5 in)
"half letter"	(5.5 x 8.5 in)
"quarto"	(8.0 x 10.0 in)
"octavo"	(6.75 x 10.5 in)
"executive"	(7.25 x 10.5 in)
"monarch"	(7.25 x 10.5 in)
"foolscap"	(8.27 x 13.0 in)
"folio"	(8.27 x 13.0 in)
"super-b"	(13.0 x 19.0 in)
"post"	(15.5 x 19.5 in)
"crown"	(15.0 x 20.0 in)
"large post"	(16.5 x 21.0 in)
"demy"	(17.5 x 22.5 in)
"medium"	(18.0 x 23.0 in)
"broadsheet"	(18.0 x 24.0 in)
"royal"	(20.0 x 25.0 in)
"elephant"	(23.0 x 28.0 in)
"double demy"	(22.5 x 35.0 in)
"quad demy"	(35.0 x 45.0 in)
"atlas"	(26.0 x 34.0 in)
"imperial"	(22.0 x 30.0 in)
"antiquarian"	(31.0 x 53.0 in)

Formats de base PA4

"pa0"	(840 x 1120 mm)
"pa1"	(560 x 840 mm)
"pa2"	(420 x 560 mm)
"pa3"	(280 x 420 mm)
"pa4"	(210 x 280 mm)
"pa5"	(140 x 210 mm)
"pa6"	(105 x 140 mm)
"pa7"	(70 x 105 mm)

"pa8" (52 x 70 mm)
 "pa9" (35 x 52 mm)
 "pa10" (26 x 35 mm)

Format utilisé en Asie du Sudest et en Australie

"f4" (210 x 330 mm)

Format spécifique aux courts exemples @lilypond de la documentation, basé sur un A8 à l'italienne.

"a8landscape" (74 x 52 mm)

A.6 Instruments MIDI

La liste suivante répertorie les différentes dénominations que vous pouvez affecter à la propriété `midiInstrument`. L'ordre dans lequel ils sont rangés, par colonne, correspond aux 128 programmes du standard *General MIDI*.

acoustic grand	contrabass	lead 7 (fifths)
bright acoustic	tremolo strings	lead 8 (bass+lead)
electric grand	pizzicato strings	pad 1 (new age)
honky-tonk	orchestral harp	pad 2 (warm)
electric piano 1	timpani	pad 3 (polysynth)
electric piano 2	string ensemble 1	pad 4 (choir)
harpsichord	string ensemble 2	pad 5 (bowed)
clav	synthstrings 1	pad 6 (metallic)
celesta	synthstrings 2	pad 7 (halo)
glockenspiel	choir aahs	pad 8 (sweep)
music box	voice oohs	fx 1 (rain)
vibraphone	synth voice	fx 2 (soundtrack)
marimba	orchestra hit	fx 3 (crystal)
xylophone	trumpet	fx 4 (atmosphere)
tubular bells	trombone	fx 5 (brightness)
dulcimer	tuba	fx 6 (goblins)
drawbar organ	muted trumpet	fx 7 (echoes)
percussive organ	french horn	fx 8 (sci-fi)
rock organ	brass section	sitar
church organ	synthbrass 1	banjo
reed organ	synthbrass 2	shamisen
accordion	soprano sax	koto
harmonica	alto sax	kalimba
concertina	tenor sax	bagpipe
acoustic guitar (nylon)	baritone sax	fiddle
acoustic guitar (steel)	oboe	shanai
electric guitar (jazz)	english horn	tinkle bell
electric guitar (clean)	bassoon	agogo
electric guitar (muted)	clarinet	steel drums
overdriven guitar	piccolo	woodblock
distorted guitar	flute	taiko drum
guitar harmonics	recorder	melodic tom
acoustic bass	pan flute	synth drum
electric bass (finger)	blown bottle	reverse cymbal
electric bass (pick)	shakuhachi	guitar fret noise
fretless bass	whistle	breath noise
slap bass 1	ocarina	seashore

slap bass 2	lead 1 (square)	bird tweet
synth bass 1	lead 2 (sawtooth)	telephone ring
synth bass 2	lead 3 (calliope)	helicopter
violin	lead 4 (chiff)	applause
viola	lead 5 (charang)	gunshot
cello	lead 6 (voice)	

A.7 Liste des couleurs

Couleurs de base

La syntaxe appropriée à la gestion des couleurs est traitée au chapitre [Coloration d'objets], page 205.

black	white	red	green
blue	cyan	magenta	yellow
grey	darkred	darkgreen	darkblue
darkcyan	darkmagenta	darkyellow	

Noms de couleur X

Les noms de couleur X peuvent s'employer de différentes manières.

Un nom de couleur peut s'écrire sous la forme d'un mot composé et sans espace auquel chaque élément prend une initiale capitalisée (p. ex. `LightSlateBlue`). Il peut aussi s'exprimer sous la forme d'une suite de mots, auquel cas les initiales ne sont pas capitalisées (par ex. `light slate blue`).

Le *gris* accepte aussi bien l'orthographe « grey » que « gray » (par ex. `DarkSlateGray`).

Certains noms peuvent prendre un suffixe numérique, comme `LightSalmon4`.

Noms de couleur sans suffixe numérique

snow	GhostWhite	WhiteSmoke	gainsboro	FloralWhite
OldLace	linen	AntiqueWhite	PapayaWhip	BlanchedAlmond
bisque	PeachPuff	NavajoWhite	moccasin	cornsilk
ivory	LemonChiffon	seashell	honeydew	MintCream
azure	AliceBlue	lavender	LavenderBlush	MistyRose
white	black	DarkSlateGrey	DimGrey	SlateGrey
LightSlateGrey	grey	LightGrey	MidnightBlue	navy
NavyBlue	CornflowerBlue	DarkSlateBlue	SlateBlue	MediumSlateBlue
LightSlateBlue	MediumBlue	RoyalBlue	blue	DodgerBlue
DeepSkyBlue	SkyBlue	LightSkyBlue	SteelBlue	LightSteelBlue
LightBlue	PowderBlue	PaleTurquoise	DarkTurquoise	MediumTurquoise
turquoise	cyan	LightCyan	CadetBlue	MediumAquamarine
aquamarine	DarkGreen	DarkOliveGreen	DarkSeaGreen	SeaGreen
MediumSeaGreen	LightSeaGreen	PaleGreen	SpringGreen	LawnGreen
green	chartreuse	MediumSpringGreen	GreenYellow	LimeGreen
YellowGreen	ForestGreen	OliveDrab	DarkKhaki	khaki
PaleGoldenrod	LightGoldenrodYellow	LightYellow	yellow	gold
LightGoldenrod	goldenrod	DarkGoldenrod	RosyBrown	IndianRed
SaddleBrown	sienna	peru	burlywood	beige
wheat	SandyBrown	tan	chocolate	firebrick
brown	DarkSalmon	salmon	LightSalmon	orange
DarkOrange	coral	LightCoral	tomato	OrangeRed
red	HotPink	DeepPink	pink	LightPink

PaleVioletRed	maroon	MediumVioletRed	VioletRed	magenta
violet	plum	orchid	MediumOrchid	DarkOrchid
DarkViolet	BlueViolet	purple	MediumPurple	thistle
DarkGrey	DarkBlue	DarkCyan	DarkMagenta	DarkRed
LightGreen				

Noms de couleur avec suffixe numérique

Les couleurs suivantes acceptent un suffixe numérique *N* compris entre 1 et 4.

snowN	seashellN	AntiqueWhiteN	bisqueN	PeachPuffN
NavajoWhiteN	LemonChiffonN	cornsilkN	ivoryN	honeydewN
LavenderBlushN	MistyRoseN	azureN	SlateBlueN	RoyalBlueN
blueN	DodgerBlueN	SteelBlueN	DeepSkyBlueN	SkyBlueN
LightSkyBlueN	LightSteelBlueN	LightBlueN	LightCyanN	PaleTurquoiseN
CadetBlueN	turquoiseN	cyanN	aquamarineN	DarkSeaGreenN
SeaGreenN	PaleGreenN	SpringGreenN	greenN	chartreuseN
OliveDrabN	DarkOliveGreenN	khakiN	LightGoldenrodN	LightYellowN
yellowN	goldN	goldenrodN	DarkGoldenrodN	RosyBrownN
IndianRedN	siennaN	burlywoodN	wheatN	tanN
chocolateN	firebrickN	brownN	salmonN	LightSalmonN
orangeN	DarkOrangeN	coralN	tomatoN	OrangeRedN
redN	DeepPinkN	HotPinkN	pinkN	LightPinkN
PaleVioletRedN	maroonN	VioletRedN	magentaN	orchidN
plumN	MediumOrchidN	DarkOrchidN	purpleN	MediumPurpleN
thistleN				

Échelle de gris

Une échelle de gris s'obtient en utilisant

`greyN`

où *N* est compris entre 0 et 100.

A.8 La fonte Feta

Voici, regroupés par type, les différents symboles de la fonte Emmentaler qui peuvent s'inclure directement dans un objet *markup*. Pour ce faire, il suffit d'employer le nom du glyphe, comme dans `g~\markup {\musicglyph #"scripts.segno" }` ou `\markup {\musicglyph #"five"}`. Pour de plus amples informations, reportez-vous au chapitre [Section 1.8.2 \[Mise en forme du texte\]](#), page 221.

Glyphes de clef

`clefs.C`



`clefs.C_change`



`clefs.F`



`clefs.F_change`



`clefs.G`



`clefs.G_change`



clefs.percussion	 	clefs.percussion_change	
clefs.tab	<i>T</i> <i>A</i> <i>B</i>	clefs.tab_change	<i>T</i> <i>A</i> <i>B</i>

Glyphes de métrique

















timesig.C44	C	timesig.C22	¢
-------------	----------	-------------	----------

Glyphes de chiffre







plus	+	comma	,
hyphen	-	period	.
zero	0	one	1
two	2	three	3
four	4	five	5
six	6	seven	7
eight	8	nine	9

Glyphes d'altération



















accidentals.sharp	#	accidentals .sharp.arrowup	↗#
accidentals .sharp.arrowdown	↘#	accidentals .sharp.arrowboth	↕#
accidentals.sharp .slashslash.stem	‡	accidentals.sharp .slashslashslash.stemstem	‡‡
accidentals.sharp .slashslashslash.stem	‡‡	accidentals.sharp .slashslash.stemstemstem	‡‡‡

<code>accidentals.natural</code>		<code>accidentals.natural.arrowup</code>	
<code>accidentals.natural.arrowdown</code>		<code>accidentals.natural.arrowboth</code>	
<code>accidentals.flat</code>		<code>accidentals.flat.arrowup</code>	
<code>accidentals.flat.arrowdown</code>		<code>accidentals.flat.arrowboth</code>	
<code>accidentals.flat.slash</code>		<code>accidentals.flat.slashslash</code>	
<code>accidentals.mirroredflat.flat</code>		<code>accidentals.mirroredflat</code>	
<code>accidentals.mirroredflat.backslash</code>		<code>accidentals.flatflat</code>	
<code>accidentals.flatflat.slash</code>		<code>accidentals.doublsharp</code>	
<code>accidentals.rightparen</code>)	<code>accidentals.leftparen</code>	(











Glyphes de tête de note par défaut

<code>noteheads.uM2</code>		<code>noteheads.dM2</code>	
<code>noteheads.sM1</code>		<code>noteheads.s0</code>	
<code>noteheads.s1</code>		<code>noteheads.s2</code>	

Glyphes de tête de note spéciale

<code>noteheads.sM1double</code>		<code>noteheads.s0diamond</code>	
<code>noteheads.s1diamond</code>		<code>noteheads.s2diamond</code>	
<code>noteheads.s0triangle</code>		<code>noteheads.d1triangle</code>	
<code>noteheads.u1triangle</code>		<code>noteheads.u2triangle</code>	
<code>noteheads.d2triangle</code>		<code>noteheads.s0slash</code>	
<code>noteheads.s1slash</code>		<code>noteheads.s2slash</code>	
<code>noteheads.s0cross</code>		<code>noteheads.s1cross</code>	
<code>noteheads.s2cross</code>		<code>noteheads.s2xcircle</code>	
<code>noteheads.s0harmonic</code>		<code>noteheads.s2harmonic</code>	






















Glyphes de tête de note à forme variable

<code>noteheads.s0do</code>		<code>noteheads.d1do</code>	
<code>noteheads.u1do</code>		<code>noteheads.d2do</code>	
<code>noteheads.u2do</code>		<code>noteheads.s0doThin</code>	
<code>noteheads.d1doThin</code>		<code>noteheads.u1doThin</code>	
<code>noteheads.d2doThin</code>		<code>noteheads.u2doThin</code>	

noteheads.s0re	◡	noteheads.u1re	◡
noteheads.d1re	◡	noteheads.u2re	◡
noteheads.d2re	◡	noteheads.s0reThin	◡
noteheads.u1reThin	◡	noteheads.d1reThin	◡
noteheads.u2reThin	◡	noteheads.d2reThin	◡
noteheads.s0mi	◊	noteheads.s1mi	◊
noteheads.s2mi	◊	noteheads.s0miMirror	◊
noteheads.s1miMirror	◊	noteheads.s2miMirror	◊
noteheads.s0miThin	◊	noteheads.s1miThin	◊
noteheads.s2miThin	◊	noteheads.u0fa	◡
noteheads.d0fa	◡	noteheads.u1fa	◡
noteheads.d1fa	◡	noteheads.u2fa	◡
noteheads.d2fa	◡	noteheads.u0faThin	◡
noteheads.d0faThin	◡	noteheads.u1faThin	◡
noteheads.d1faThin	◡	noteheads.u2faThin	◡


noteheads.d2faThin	▴	noteheads.s0sol	◌
noteheads.s1sol	◌	noteheads.s2sol	●
noteheads.s0la	▢	noteheads.s1la	▢
noteheads.s2la	■	noteheads.s0laThin	▢
noteheads.s1laThin	▢	noteheads.s2laThin	■
noteheads.s0ti	◊	noteheads.u1ti	◊
noteheads.d1ti	◊	noteheads.u2ti	◆
noteheads.d2ti	◆	noteheads.s0tiThin	◊
noteheads.u1tiThin	◊	noteheads.d1tiThin	◊
noteheads.u2tiThin	◆	noteheads.d2tiThin	◆
noteheads.u0doFunk	▷	noteheads.d0doFunk	▷
noteheads.u1doFunk	▷	noteheads.d1doFunk	▷
noteheads.u2doFunk	▴	noteheads.d2doFunk	▴
noteheads.u0reFunk	▷	noteheads.d0reFunk	▷
noteheads.u1reFunk	▷	noteheads.d1reFunk	▷

noteheads.u2reFunk	►	noteheads.d2reFunk	◄
noteheads.u0miFunk	◊	noteheads.d0miFunk	◊
noteheads.u1miFunk	◊	noteheads.d1miFunk	◊
noteheads.s2miFunk	◆	noteheads.u0faFunk	▼
noteheads.d0faFunk	▴	noteheads.u1faFunk	▼
noteheads.d1faFunk	▴	noteheads.u2faFunk	▼
noteheads.d2faFunk	▴	noteheads.s0solFunk	○
noteheads.s1solFunk	○	noteheads.s2solFunk	●
noteheads.s0laFunk	□	noteheads.s1laFunk	□
noteheads.s2laFunk	■	noteheads.u0tiFunk	▷
noteheads.d0tiFunk	◁	noteheads.u1tiFunk	▷
noteheads.d1tiFunk	◁	noteheads.u2tiFunk	►
noteheads.d2tiFunk	◄	noteheads.s0doWalker	△
noteheads.u1doWalker	▽	noteheads.d1doWalker	△
noteheads.u2doWalker	▼	noteheads.d2doWalker	▲













<code>noteheads.s0reWalker</code>		<code>noteheads.u1reWalker</code>	
<code>noteheads.d1reWalker</code>		<code>noteheads.u2reWalker</code>	
<code>noteheads.d2reWalker</code>		<code>noteheads.s0miWalker</code>	
<code>noteheads.s1miWalker</code>		<code>noteheads.s2miWalker</code>	
<code>noteheads.s0faWalker</code>		<code>noteheads.u1faWalker</code>	
<code>noteheads.d1faWalker</code>		<code>noteheads.u2faWalker</code>	
<code>noteheads.d2faWalker</code>		<code>noteheads.s0laWalker</code>	
<code>noteheads.s1laWalker</code>		<code>noteheads.s2laWalker</code>	
<code>noteheads.s0tiWalker</code>		<code>noteheads.ultiWalker</code>	
<code>noteheads.d1tiWalker</code>		<code>noteheads.u2tiWalker</code>	
<code>noteheads.d2tiWalker</code>			

Glyphes de silence

<code>rests.0</code>		<code>rests.1</code>	
<code>rests.0o</code>		<code>rests.1o</code>	
<code>rests.M3</code>		<code>rests.M2</code>	
<code>rests.M1</code>		<code>rests.M1o</code>	

rests.2		rests.2classical	
rests.3		rests.4	
rests.5		rests.6	
rests.7			

Glyphes de crochet de croche

flags.u3		flags.u4	
flags.u5		flags.u6	
flags.u7		flags.d3	
flags.d4		flags.d5	
flags.d6		flags.d7	
flags.ugrace		flags.dgrace	

Glyphes de point

dots.dot	
----------	---




















Glyphes de nuance

space		f	<i>f</i>
m	<i>m</i>	p	<i>p</i>
r	<i>r</i>	s	<i>s</i>
z	<i>z</i>		









Glyphes de script

scripts.ufermata		scripts.dfermata	
scripts.ushortfermata		scripts.dshortfermata	
scripts.ulongfermata		scripts.dlongfermata	
scripts.uverylongfermata		scripts.dverylongfermata	
scripts.thumb		scripts.sforzato	
scripts.espr		scripts.staccato	
scripts.ustaccatissimo		scripts.dstaccatissimo	
scripts.tenuto		scripts.uportato	
scripts.dportato		scripts.umarcato	
scripts.dmarcato		scripts.open	



<code>scripts.halfopen</code>		<code>scripts.halfopenvertical</code>	
<code>scripts.stopped</code>		<code>scripts.upbow</code>	
<code>scripts.downbow</code>		<code>scripts.reverseturn</code>	
<code>scripts.turn</code>		<code>scripts.trill</code>	
<code>scripts.upedalheel</code>		<code>scripts.dpedalheel</code>	
<code>scripts.upedaltoe</code>		<code>scripts.dpedaltoe</code>	
<code>scripts.flageolet</code>		<code>scripts.segno</code>	
<code>scripts.varsegno</code>		<code>scripts.coda</code>	
<code>scripts.varcoda</code>		<code>scripts.rcomma</code>	
<code>scripts.lcomma</code>		<code>scripts.rvarcomma</code>	
<code>scripts.lvarcomma</code>		<code>scripts.arpeggio</code>	
<code>scripts.trill_element</code>		<code>scripts.arpeggio .arrow.M1</code>	
<code>scripts.arpeggio.arrow.1</code>		<code>scripts.trilelement</code>	
<code>scripts.prall</code>		<code>scripts.mordent</code>	
<code>scripts.prallprall</code>		<code>scripts.prallmordent</code>	

<code>scripts.upprall</code>		<code>scripts.upmordent</code>	
<code>scripts.pralldown</code>		<code>scripts.downprall</code>	
<code>scripts.downmordent</code>		<code>scripts.prallup</code>	
<code>scripts.lineprall</code>		<code>scripts.caesura.curved</code>	
<code>scripts.caesura.straight</code>		<code>scripts.snappizzicato</code>	
<code>scripts.ictus</code>		<code>scripts.uaccentus</code>	
<code>scripts.daccentus</code>		<code>scripts.usemicirculus</code>	
<code>scripts.dsemicirculus</code>		<code>scripts.circulus</code>	
<code>scripts.augmentum</code>		<code>scripts .usignumcongruentiae</code>	
<code>scripts .dsignumcongruentiae</code>			

Glyphes de flèche

<code>arrowheads.open.01</code>		<code>arrowheads.open.0M1</code>	
<code>arrowheads.open.11</code>		<code>arrowheads.open.1M1</code>	
<code>arrowheads.close.01</code>		<code>arrowheads.close.0M1</code>	
<code>arrowheads.close.11</code>		<code>arrowheads.close.1M1</code>	


Glyphes d'extrémité d'accolade

<code>brackettips.up</code>		<code>brackettips.down</code>	
-----------------------------	---	-------------------------------	---

Glyphes de pédale



<code>pedal.*</code>		<code>pedal.M</code>	
----------------------	---	----------------------	---



<code>pedal..</code>		<code>pedal.P</code>	
----------------------	---	----------------------	---

<code>pedal.d</code>		<code>pedal.e</code>	
----------------------	---	----------------------	---

<code>pedal.Ped</code>	
------------------------	---

Glyphes d'accordéon

<code>accordion.discant</code>		<code>accordion.dot</code>	
--------------------------------	---	----------------------------	---

<code>accordion.freebass</code>		<code>accordion.stdbass</code>	
---------------------------------	---	--------------------------------	---

<code>accordion.bayanbass</code>		<code>accordion.oldEE</code>	
----------------------------------	---	------------------------------	---

<code>accordion.push</code>		<code>accordion.pull</code>	
-----------------------------	---	-----------------------------	---
















Glyphes de liaison

<code>ties.lyric.short</code>		<code>ties.lyric.default</code>	
-------------------------------	---	---------------------------------	---

Glyphes de style vaticana


<code>clefs.vaticana.do</code>		<code>clefs.vaticana.do_change</code>	
<code>clefs.vaticana.fa</code>		<code>clefs.vaticana.fa_change</code>	
<code>custodes.vaticana.u0</code>		<code>custodes.vaticana.u1</code>	
<code>custodes.vaticana.u2</code>		<code>custodes.vaticana.d0</code>	
<code>custodes.vaticana.d1</code>		<code>custodes.vaticana.d2</code>	
<code>accidentals.vaticanaM1</code>		<code>accidentals.vaticana0</code>	
<code>dots.dotvaticana</code>		<code>noteheads</code> <code>.svaticana.punctum</code>	
<code>noteheads.svaticana</code> <code>.punctum.cavum</code>		<code>noteheads.svaticana</code> <code>.linea.punctum</code>	
<code>noteheads.svaticana</code> <code>.linea.punctum.cavum</code>		<code>noteheads.svaticana</code> <code>.inclinatum</code>	
<code>noteheads.svaticana.lpes</code>		<code>noteheads</code> <code>.svaticana.vlpes</code>	
<code>noteheads.svaticana.upes</code>		<code>noteheads</code> <code>.svaticana.vupes</code>	
<code>noteheads</code> <code>.svaticana.plica</code>		<code>noteheads</code> <code>.svaticana.vplica</code>	
<code>noteheads</code> <code>.svaticana.epiphonus</code>		<code>noteheads.svaticana</code> <code>.vepiphonus</code>	
<code>noteheads.svaticana</code> <code>.reverse.plica</code>		<code>noteheads.svaticana</code> <code>.reverse.vplica</code>	
<code>noteheads.svaticana</code> <code>.inner.cephalicus</code>		<code>noteheads.svaticana</code> <code>.cephalicus</code>	
<code>noteheads</code> <code>.svaticana.quilisma</code>			

Glyphes de style medicaea

<code>clefs.medicaea.do</code>		<code>clefs.medicaea.do_change</code>	
<code>clefs.medicaea.fa</code>		<code>clefs.medicaea.fa_change</code>	
<code>custodes.medicaea.u0</code>		<code>custodes.medicaea.u1</code>	
<code>custodes.medicaea.u2</code>		<code>custodes.medicaea.d0</code>	
<code>custodes.medicaea.d1</code>		<code>custodes.medicaea.d2</code>	
<code>accidentals.medicaeaM1</code>		<code>noteheads.smedicaea.inclinatum</code>	
<code>noteheads.smedicaea.punctum</code>		<code>noteheads.smedicaea.rvirga</code>	
<code>noteheads.smedicaea.virga</code>			

Glyphes de style Hufnagel

<code>clefs.hufnagel.do</code>		<code>clefs.hufnagel.do_change</code>	
<code>clefs.hufnagel.fa</code>		<code>clefs.hufnagel.fa_change</code>	
<code>clefs.hufnagel.do.fa</code>		<code>clefs.hufnagel.do.fa_change</code>	
<code>custodes.hufnagel.u0</code>		<code>custodes.hufnagel.u1</code>	
<code>custodes.hufnagel.u2</code>		<code>custodes.hufnagel.d0</code>	
<code>custodes.hufnagel.d1</code>		<code>custodes.hufnagel.d2</code>	

accidentals.hufnagelM1		noteheads .shufnagel.punctum	
------------------------	---	---------------------------------	---

noteheads .shufnagel.virga		noteheads.shufnagel.lpes	
-------------------------------	---	--------------------------	---

Glyphes de style mensural


rests.M3mensural		rests.M2mensural	
------------------	---	------------------	---

rests.M1mensural		rests.0mensural	
------------------	---	-----------------	---

rests.1mensural		rests.2mensural	
-----------------	---	-----------------	---

rests.3mensural		rests.4mensural	
-----------------	---	-----------------	---

clefs.mensural.c		clefs.mensural.c_change	
------------------	---	-------------------------	---

clefs.mensural.f		clefs.mensural.f_change	
------------------	---	-------------------------	---


clefs.mensural.g		clefs.mensural.g_change	
------------------	---	-------------------------	---

custodes.mensural.u0		custodes.mensural.u1	
----------------------	---	----------------------	---































custodes.mensural.u2		custodes.mensural.d0	
----------------------	---	----------------------	---















custodes.mensural.d1		custodes.mensural.d2	
----------------------	---	----------------------	---

accidentals.mensural1		accidentals.mensuralM1	
-----------------------	---	------------------------	---

















flags.mensuralu03		flags.mensuralu13	
-------------------	---	-------------------	---



















flags.mensuralu23)	flags.mensurald03	(
flags.mensurald13	(flags.mensurald23	(
flags.mensuralu04	}	flags.mensuralu14	}
flags.mensuralu24	}	flags.mensurald04	{
flags.mensurald14	{	flags.mensurald24	{
flags.mensuralu05	}	flags.mensuralu15	}
flags.mensuralu25	}	flags.mensurald05	{
flags.mensurald15	{	flags.mensurald25	{
flags.mensuralu06	}	flags.mensuralu16	}
flags.mensuralu26	}	flags.mensurald06	{
flags.mensurald16	{	flags.mensurald26	{
timesig.mensural44	C	timesig.mensural22	¢
timesig.mensural32	O	timesig.mensural64	©
timesig.mensural94	⊙	timesig.mensural34	⊕

timesig.mensural68		timesig.mensural98	
timesig.mensural48		timesig.mensural68alt	
timesig.mensural24		noteheads.um3mensural	
noteheads.dm3mensural		noteheads.sm3ligmensural	
noteheads.um2mensural		noteheads.dm2mensural	
noteheads.sm2ligmensural		noteheads.sm1mensural	
noteheads.ur3mensural		noteheads.dr3mensural	
noteheads .sr3ligmensural		noteheads.ur2mensural	
noteheads.dr2mensural		noteheads .sr2ligmensural	
noteheads.sr1mensural		noteheads .um3semimensural	
noteheads .dm3semimensural		noteheads .sm3semiligmensural	
noteheads .um2semimensural		noteheads .dm2semimensural	
noteheads .sm2semiligmensural		noteheads .sm1semimensural	
noteheads .ur3semimensural		noteheads .dr3semimensural	
noteheads .sr3semiligmensural		noteheads .ur2semimensural	











noteheads .drM2semimensural		noteheads .srM2semiligmensural	
noteheads .srM1semimensural		noteheads .uM3blackmensural	
noteheads .dM3blackmensural		noteheads .sM3blackligmensural	
noteheads .uM2blackmensural		noteheads .dM2blackmensural	
noteheads .sM2blackligmensural		noteheads .sM1blackmensural	
noteheads.s0mensural		noteheads.s1mensural	
noteheads.s2mensural		noteheads .s0blackmensural	











Glyphes de style néomensural

rests.M3neomensural		rests.M2neomensural	
rests.M1neomensural		rests.0neomensural	
rests.1neomensural		rests.2neomensural	
rests.3neomensural		rests.4neomensural	
clefs.neomensural.c		clefs.neomensural .c_change	
timesig.neomensural44		timesig.neomensural22	
timesig.neomensural32		timesig.neomensural64	
timesig.neomensural94		timesig.neomensural34	








timesig.neomensural68		timesig.neomensural98	
timesig.neomensural48		timesig.neomensural68alt	
timesig.neomensural24		noteheads.um3neomensural	
noteheads.dm3neomensural		noteheads.um2neomensural	
noteheads.dm2neomensural		noteheads.sm1neomensural	
noteheads.urM3neomensural		noteheads.drM3neomensural	
noteheads.urM2neomensural		noteheads.drM2neomensural	
noteheads.srM1neomensural		noteheads.s0neomensural	
noteheads.slneomensural		noteheads.s2neomensural	

Glyphes de style Petrucci

clefs.petrucci.c1		clefs.petrucci.c1_change	
clefs.petrucci.c2		clefs.petrucci.c2_change	
clefs.petrucci.c3		clefs.petrucci.c3_change	
clefs.petrucci.c4		clefs.petrucci.c4_change	
clefs.petrucci.c5		clefs.petrucci.c5_change	






<code>clefs.petrucchi.f</code>		<code>clefs.petrucchi.f_change</code>	
<code>clefs.petrucchi.g</code>		<code>clefs.petrucchi.g_change</code>	
<code>noteheads.s0petrucci</code>		<code>noteheads.s1petrucci</code>	
<code>noteheads.s2petrucci</code>		<code>noteheads.s0blackpetrucci</code>	
<code>noteheads.s1blackpetrucci</code>		<code>noteheads.s2blackpetrucci</code>	

Glyphes de style Solesmes

<code>noteheads.ssolesmes.incl.parvum</code>		<code>noteheads.ssolesmes.auct.asc</code>	
<code>noteheads.ssolesmes.auct.desc</code>		<code>noteheads.ssolesmes.incl.auctum</code>	
<code>noteheads.ssolesmes.stropha</code>		<code>noteheads.ssolesmes.stropha.aucta</code>	
<code>noteheads.ssolesmes.oriscus</code>			

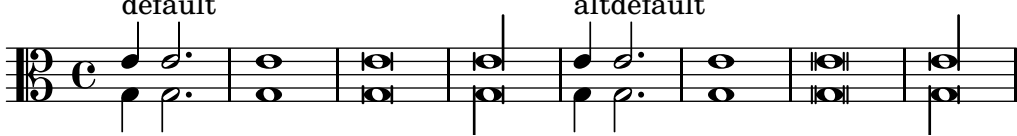
Glyphes de style kiévien

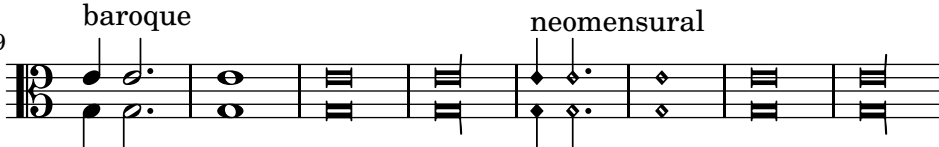
<code>clefs.kievan.do</code>		<code>clefs.kievan.do_change</code>	
<code>accidentals.kievan1</code>		<code>accidentals.kievanM1</code>	
<code>scripts.barline.kievan</code>		<code>dots.dotkievan</code>	
<code>noteheads.sM2kievan</code>		<code>noteheads.sM1kievan</code>	
<code>noteheads.s0kievan</code>		<code>noteheads.d2kievan</code>	

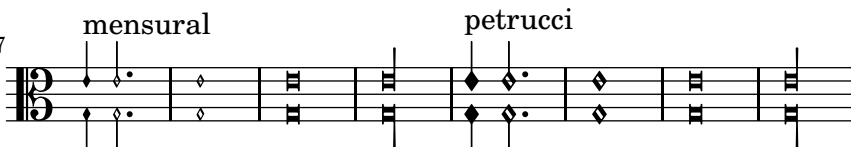
<code>noteheads.u2kievan</code>		<code>noteheads.slkievan</code>	
<code>noteheads.sr1kievan</code>		<code>noteheads.d3kievan</code>	
<code>noteheads.u3kievan</code>			

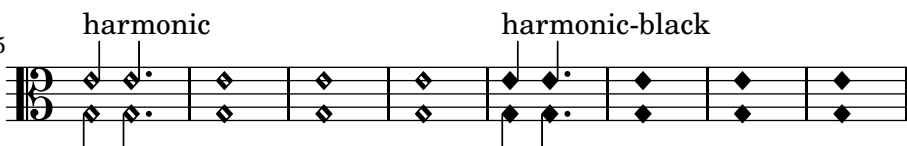
A.9 Styles de tête de note

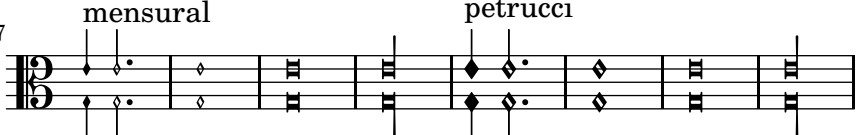
Voici les différents styles de tête de note disponibles.

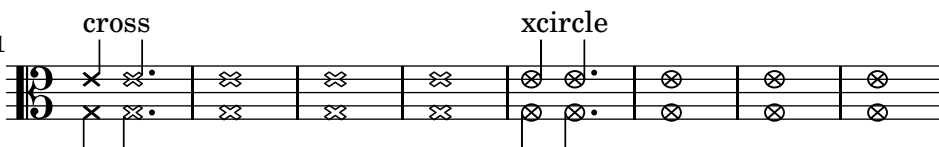
default 

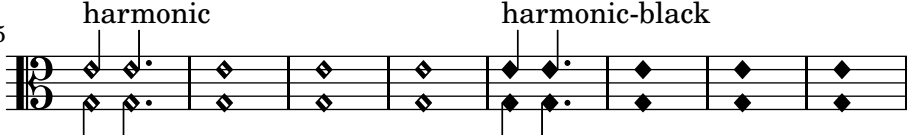
altdefault 

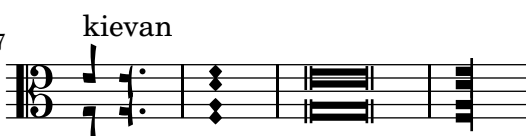
9 baroque 


neomensural 

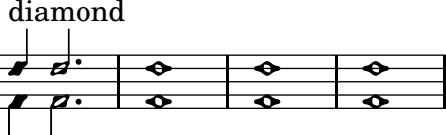
17 mensural 

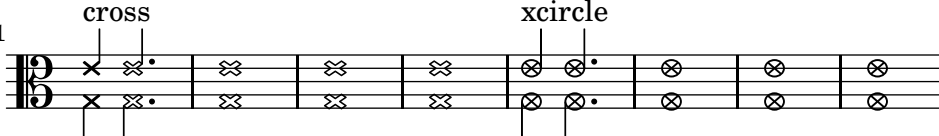
petrucci 

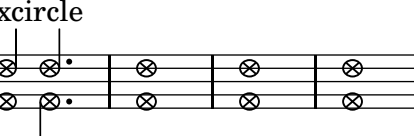
25 harmonic 

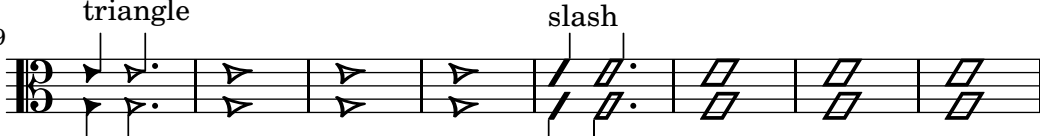
harmonic-black 

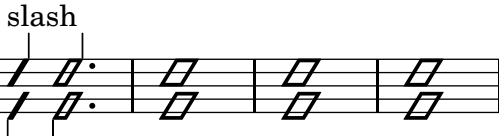
33 harmonic-mixed 

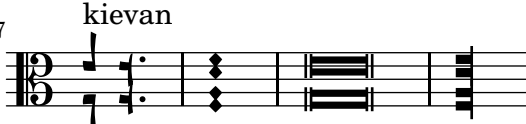
diamond 

41 cross 

xcircle 

49 triangle 

slash 

57 kievian 

A.10 Commandes pour markup

Les commandes suivantes peuvent être utilisées dans un bloc `\markup { }`.

The following commands can all be used inside `\markup { }`.

A.10.1 Font

`\abs-fontsize` *size* (number) *arg* (markup)

Use *size* as the absolute font size to display *arg*. Adjusts `baseline-skip` and `word-space` accordingly.

```
\markup {
  default text font size
  \hspace #2
  \abs-fontsize #16 { text font size 16 }
  \hspace #2
  \abs-fontsize #12 { text font size 12 }
}
```

default text font size **text font size 16** text font size 12

`\bold` *arg* (markup)

Switch to bold font-series.

```
\markup {
  default
  \hspace #2
  \bold
  bold
}
```

default **bold**

`\box` *arg* (markup)

Draw a box round *arg*. Looks at `thickness`, `box-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \override #'(box-padding . 0.5)
  \box
  \line { V. S. }
}
```

V. S.

Used properties:

- `box-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\caps` *arg* (markup)

Copy of the `\smallCaps` command.

```
\markup {
  default
  \hspace #2
```

```

\caps {
  Text in small caps
}

```

default **TEXT IN SMALL CAPS**

\dynamic *arg* (markup)

Use the dynamic font. This font only contains **s**, **f**, **m**, **z**, **p**, and **r**. When producing phrases, like ‘più **f**’, the normal words (like ‘più’) should be done in a different font. The recommended font for this is bold and italic.

```

\markup {
  \dynamic {
    sfzp
  }
}

```

sfzp

\finger *arg* (markup)

Set *arg* as small numbers.

```

\markup {
  \finger {
    1 2 3 4 5
  }
}

```

1 2 3 4 5

\fontCaps *arg* (markup)

Set **font-shape** to caps

Note: **\fontCaps** requires the installation and selection of fonts which support the caps font shape.

\fontsize *increment* (number) *arg* (markup)

Add *increment* to the font-size. Adjusts **baseline-skip** accordingly.

```

\markup {
  default
  \hspace #2
  \fontsize #-1.5
  smaller
}

```

default **smaller**

Used properties:

- **baseline-skip** (2)
- **word-space** (1)
- **font-size** (0)

\huge *arg* (markup)

Set font size to +2.

```
\markup {
  default
  \hspace #2
  \huge
  huge
}
```

default huge

`\italic arg` (markup)
Use italic font-shape for *arg*.

```
\markup {
  default
  \hspace #2
  \italic
  italic
}
```

default *italic*

`\large arg` (markup)
Set font size to +1.

```
\markup {
  default
  \hspace #2
  \large
  large
}
```

default large

`\larger arg` (markup)
Increase the font size relative to the current setting.

```
\markup {
  default
  \hspace #2
  \larger
  larger
}
```

default larger

`\magnify sz` (number) *arg* (markup)
Set the font magnification for its argument. In the following example, the middle A is 10% larger:

```
A \magnify #1.1 { A } A
```

Note: Magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.

```
\markup {
  default
  \hspace #2
```

```

\magnify #1.5 {
  50% larger
}

```

default **50% larger**

`\medium arg` (markup)

Switch to medium font-series (in contrast to bold).

```

\markup {
  \bold {
    some bold text
    \hspace #2
    \medium {
      medium font series
    }
    \hspace #2
    bold again
  }
}

```

some bold text medium font series **bold again**

`\normal-size-sub arg` (markup)

Set *arg* in subscript with a normal font size.

```

\markup {
  default
  \normal-size-sub {
    subscript in standard size
  }
}

```

default subscript in standard size

Used properties:

- baseline-skip

`\normal-size-super arg` (markup)

Set *arg* in superscript with a normal font size.

```

\markup {
  default
  \normal-size-super {
    superscript in standard size
  }
}

```

default superscript in standard size

Used properties:

- baseline-skip

`\normal-text` *arg* (markup)

Set all font related properties (except the size) to get the default normal text font, no matter what font was used earlier.

```
\markup {
  \huge \bold \sans \caps {
    huge bold sans caps
  }
  \hspace #2
  \normal-text {
    huge normal
  }
  \hspace #2
  as before
}
```

HUGE BOLD SANS CAPS huge normal **AS BEFORE**

`\normalsize` *arg* (markup)

Set font size to default.

```
\markup {
  \teeny {
    this is very small
  }
  \hspace #2
  \normalsize {
    normal size
  }
  \hspace #2
  teeny again
}
```

this is very small **normal size** teeny again

`\number` *arg* (markup)

Set font family to **number**, which yields the font used for time signatures and fingerings. This font contains numbers and some punctuation; it has no letters.

```
\markup {
  \number {
    0 1 2 3 4 5 6 7 8 9 . ,
  }
}
```

0123456789.,

`\replace` *replacements* (list) *arg* (markup)

Used to automatically replace a string by another in the markup *arg*. Each pair of the alist *replacements* specifies what should be replaced. The **key** is the string to be replaced by the **value** string.

```
\markup \replace #'(("thx" . "Thanks!")) thx
```

Thanks!


```
\roman arg (markup)
  Set font family to roman.
  \markup {
    \sans \bold {
      sans serif, bold
      \hspace #2
      \roman {
        text in roman font family
      }
      \hspace #2
      return to sans
    }
  }
```

sans serif, bold text in roman font family return to sans

```
\sans arg (markup)
  Switch to the sans serif font family.
  \markup {
    default
    \hspace #2
    \sans {
      sans serif
    }
  }
```

default sans serif

```
\simple str (string)
  A simple text string; \markup { foo } is equivalent with \markup { \simple #"foo"
  }.
```

Note: for creating standard text markup or defining new markup commands, the use of \simple is unnecessary.

```
\markup {
  \simple #"simple"
  \simple #"text"
  \simple #"strings"
}
```

simple text strings

```
\small arg (markup)
  Set font size to -1.
  \markup {
    default
    \hspace #2
    \small
    small
  }
```

default small

`\smallCaps` *arg* (markup)

Emit *arg* as small caps.

Note: `\smallCaps` does not support accented characters.

```
\markup {
  default
  \hspace #2
  \smallCaps {
    Text in small caps
  }
}
```

default TEXT IN SMALL CAPS

`\smaller` *arg* (markup)

Decrease the font size relative to the current setting.

```
\markup {
  \fontsize #3.5 {
    some large text
    \hspace #2
    \smaller {
      a bit smaller
    }
  }
  \hspace #2
  more large text
}
```

some large text a bit smaller more large text

`\sub` *arg* (markup)

Set *arg* in subscript.

```
\markup {
  \concat {
    H
    \sub {
      2
    }
    0
  }
}
```

H₂O

Used properties:

- `baseline-skip`
- `font-size` (0)

`\super` *arg* (markup)

Set *arg* in superscript.

```
\markup {
  E =
  \concat {
    mc
    \super
    2
  }
}
```

$$E = mc^2$$

Used properties:

- `baseline-skip`
- `font-size (0)`

`\teeny arg` (markup)
Set font size to -3.

```
\markup {
  default
  \hspace #2
  \teeny
  teeny
}
```

default *teeny*

`\text arg` (markup)
Use a text font instead of music symbol or music alphabet font.

```
\markup {
  \number {
    1, 2,
    \text {
      three, four,
    }
    5
  }
}
```

1, 2, three, four, **5**

`\tiny arg` (markup)
Set font size to -2.

```
\markup {
  default
  \hspace #2
  \tiny
  tiny
}
```

default *tiny*

`\typewriter arg` (markup)
Use `font-family typewriter` for *arg*.

```
\markup {
  default
  \hspace #2
  \typewriter
  typewriter
}
```

default typewriter

`\underline arg` (markup)

Underline *arg*. Looks at `thickness` to determine line thickness, and `offset` to determine line y-offset.

```
\markup \fill-line {
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(thickness . 1)
  \underline "underlined"
  \override #'(offset . 1)
  \override #'(thickness . 5)
  \underline "underlined"
}
```

underlined

underlined

underlined

Used properties:

- `offset` (2)
- `thickness` (1)

`\upright arg` (markup)

Set `font-shape` to `upright`. This is the opposite of *italic*.

```
\markup {
  \italic {
    italic text
    \hspace #2
    \upright {
      upright text
    }
    \hspace #2
    italic again
  }
}
```

italic text upright text *italic again*

A.10.2 Align

`\center-align arg` (markup)

Align *arg* to its X center.

```
\markup {
  \column {
    one
    \center-align
```

```

        two
        three
    }
}

```

```

        one
    two
        three

```

`\center-column` *args* (markup list)

Put *args* in a centered column.

```

\markup {
    \center-column {
        one
        two
        three
    }
}

```

```

        one
        two
        three

```

Used properties:

- `baseline-skip`

`\column` *args* (markup list)

Stack the markups in *args* vertically. The property `baseline-skip` determines the space between markups in *args*.

```

\markup {
    \column {
        one
        two
        three
    }
}

```

```

        one
        two
        three

```

Used properties:

- `baseline-skip`

`\combine` *arg1* (markup) *arg2* (markup)

Print two markups on top of each other.

Note: `\combine` cannot take a list of markups enclosed in curly braces as an argument; the follow example will not compile:

```

\combine { a list }
\markup {
    \fontsize #5
    \override #'(thickness . 2)

```

```

\combine
\draw-line #'(0 . 4)
\arrow-head #Y #DOWN ##f
}

```



`\concat args` (markup list)

Concatenate *args* in a horizontal line, without spaces in between. Strings and simple markups are concatenated on the input level, allowing ligatures. For example, `\concat { "f" \simple #"i" }` is equivalent to "fi".

```

\markup {
\concat {
one
two
three
}
}

```

onetwothree

`\dir-column args` (markup list)

Make a column of *args*, going up or down, depending on the setting of the `direction` layout property.

```

\markup {
\override #`(direction . ,UP) {
\dir-column {
going up
}
}
\hspace #1
\dir-column {
going down
}
\hspace #1
\override #'(direction . 1) {
\dir-column {
going up
}
}
}

```

```

up      up
going going going
down

```

Used properties:

- `baseline-skip`
- `direction`

`\fill-line args` (markup list)

Put *markups* in a horizontal line of width *line-width*. The markups are spaced or flushed to fill the entire line. If there are no arguments, return an empty stencil.

```

\markup {
  \column {
    \fill-line {
      Words evenly spaced across the page
    }
    \null
    \fill-line {
      \line { Text markups }
      \line {
        \italic { evenly spaced }
      }
      \line { across the page }
    }
  }
}

```

Words	evenly	spaced	across	the	page
Text markups		<i>evenly spaced</i>		across the page	

Used properties:

- `line-width` (#f)
- `word-space` (0.6)
- `text-direction` (1)

`\fill-with-pattern` *space* (number) *dir* (direction) *pattern* (markup) *left* (markup) *right* (markup)

Put *left* and *right* in a horizontal line of width `line-width` with a line of markups *pattern* in between. Patterns are spaced apart by *space*. Patterns are aligned to the *dir* markup.

```

\markup \column {
  "right-aligned :"
  \fill-with-pattern #1 #RIGHT . first right
  \fill-with-pattern #1 #RIGHT . second right
  \null
  "center-aligned :"
  \fill-with-pattern #1.5 #CENTER - left right
  \null
  "left-aligned :"
  \override #'(line-width . 50)
  \fill-with-pattern #2 #LEFT : left first
  \override #'(line-width . 50)
  \fill-with-pattern #2 #LEFT : left second
}

```

right-aligned :

first right
 second right

center-aligned :

left - - - - - right

left-aligned :

left: : : : : : : : : : : : : : : first
 left: : : : : : : : : : : : : : : second

Used properties:

- line-width
- word-space

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)
 Align *arg* in *axis* direction to the *dir* side.

```
\markup {
  \column {
    one
    \general-align #X #LEFT
    two
    three
    \null
    one
    \general-align #X #CENTER
    two
    three
    \null
    \line {
      one
      \general-align #Y #UP
      two
      three
    }
    \null
    \line {
      one
      \general-align #Y #3.2
      two
      three
    }
  }
}
```



```

one
two
three

```

```

one
two
three

```

```

one   three
  two

```

```

one   three
      two

```

`\halign` *dir* (number) *arg* (markup)

Set horizontal alignment. If *dir* is `-1`, then it is left-aligned, while `+1` is right. Values in between interpolate alignment accordingly.

```

\markup {
  \column {
    one
    \halign #LEFT
    two
    three
    \null
    one
    \halign #CENTER
    two
    three
    \null
    one
    \halign #RIGHT
    two
    three
    \null
    one
    \halign #-5
    two
    three
  }
}

```

one
two
three

one
two
three

one
two
three

one
two
three

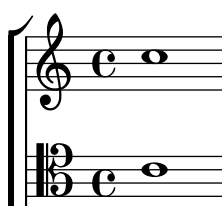
`\hcenter-in` *length* (number) *arg* (markup)

Center *arg* horizontally within a box of extending *length*/2 to the left and right.

```
\new StaffGroup <<
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Oboe
    }
    c''1
  }
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Bassoon
    }
    \clef tenor
    c'1
  }
>>
```

Oboe

Bassoon



`\hspace` *amount* (number)

Create an invisible object taking up horizontal space *amount*.

```
\markup {
  one
  \hspace #2
  two
  \hspace #8
  three
}
```

one two three

Used properties:

- `word-space`

`\justify-field` *symbol* (symbol)

Justify the data which has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur adipisicing
    elit, sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \justify-field #'header:myText
    }
  }
}

\markup {
  \null
}
```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\justify` *args* (markup list)

Like `\wordwrap`, but with lines stretched to justify the margins. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```
\markup {
  \justify {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\justify-string` *arg* (string)

Justify a string. Paragraphs may be separated with double newlines

```
\markup {
  \override #'(line-width . 40)
  \justify-string #"Lorem ipsum dolor sit amet, consectetur
    adipisicing elit, sed do eiusmod tempor incididunt ut labore
    et dolore magna aliqua.
```

Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat.

Excepteur sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum"

```
}
```

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et
dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.

Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia
deserunt mollit anim id est laborum

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

`\left-align` *arg* (markup)

Align *arg* on its left edge.

```
\markup {
  \column {
    one
    \left-align
    two
    three
  }
}
```

one
two
three

`\left-column` *args* (markup list)

Put *args* in a left-aligned column.

```
\markup {
  \left-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\line` *args* (markup list)

Put *args* in a horizontal line. The property `word-space` determines the space between markups in *args*.

```
\markup {
  \line {
    one two three
  }
}
```

one two three

Used properties:

- `text-direction` (1)
- `word-space`

`\lower` *amount* (number) *arg* (markup)

Lower *arg* by the distance *amount*. A negative *amount* indicates raising; see also `\raise`.

```
\markup {
  one
  \lower #3
  two
  three
}
```

one three
 two

`\pad-around` *amount* (number) *arg* (markup)

Add padding *amount* all around *arg*.

```
\markup {
  \box {
    default
  }
}
```

```

    }
    \hspace #2
    \box {
      \pad-around #0.5 {
        padded
      }
    }
  }
}

```

default	padded
---------	--------

`\pad-markup` *amount* (number) *arg* (markup)
Add space around a markup object.

```

\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-markup #1 {
      padded
    }
  }
}

```

default	padded
---------	--------

`\pad-to-box` *x-ext* (pair of numbers) *y-ext* (pair of numbers) *arg* (markup)
Make *arg* take at least *x-ext*, *y-ext* space.

```

\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-to-box #'(0 . 10) #'(0 . 3) {
      padded
    }
  }
}

```

default	padded
---------	--------

`\pad-x` *amount* (number) *arg* (markup)
Add padding *amount* around *arg* in the X direction.

```

\markup {
  \box {
    default
  }
}

```

```

\hspace #4
\box {
  \pad-x #2 {
    padded
  }
}
}

```

default	padded
---------	--------

`\put-adjacent` *axis* (integer) *dir* (direction) *arg1* (markup) *arg2* (markup)

Put *arg2* next to *arg1*, without moving *arg1*.

`\raise` *amount* (number) *arg* (markup)

Raise *arg* by the distance *amount*. A negative *amount* indicates lowering, see also `\lower`.

The argument to `\raise` is the vertical displacement amount, measured in (global) staff spaces. `\raise` and `\super` raise objects in relation to their surrounding markups.

If the text object itself is positioned above or below the staff, then `\raise` cannot be used to move it, since the mechanism that positions it next to the staff cancels any shift made with `\raise`. For vertical positioning, use the `padding` and/or `extra-offset` properties.

```

\markup {
  C
  \small
  \bold
  \raise #1.0
  9/7+
}

```

C 9/7+

`\right-align` *arg* (markup)

Align *arg* on its right edge.

```

\markup {
  \column {
    one
    \right-align
    two
    three
  }
}

```

one
two
three

`\right-column` *args* (markup list)

Put *args* in a right-aligned column.

```
\markup {
  \right-column {
    one
    two
    three
  }
}
```

```
one
two
three
```

Used properties:

- `baseline-skip`

`\rotate` *ang* (number) *arg* (markup)

Rotate object with *ang* degrees around its center.

```
\markup {
  default
  \hspace #2
  \rotate #45
  \line {
    rotated 45°
  }
}
```

```
default
```

```
rotated 45°
```

`\translate` *offset* (pair of numbers) *arg* (markup)

Translate *arg* relative to its surroundings. *offset* is a pair of numbers representing the displacement in the X and Y axis.

```
\markup {
  *
  \translate #'(2 . 3)
  \line { translated two spaces right, three up }
}
```

```
translated two spaces right, three up
```

```
*
```

`\translate-scaled` *offset* (pair of numbers) *arg* (markup)

Translate *arg* by *offset*, scaling the offset by the `font-size`.

```
\markup {
  \fontsize #5 {
    * \translate #'(2 . 3) translate
    \hspace #2
    * \translate-scaled #'(2 . 3) translate-scaled
  }
}
```


\ast **translate** \ast **translate-scaled**

Used properties:

- `font-size` (0)

`\vcenter` *arg* (markup)

Align *arg* to its Y center.

```
\markup {
  one
  \vcenter
  two
  three
}
```

one two three

`\vspace` *amount* (number)

Create an invisible object taking up vertical space of *amount* multiplied by 3.

```
\markup {
  \center-column {
    one
    \vspace #2
    two
    \vspace #5
    three
  }
}
```

one

two

three

`\wordwrap-field` *symbol* (symbol)

Wordwrap the data which has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur adipisicing
    elit, sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea commodo
    consequat."
}
```

```
\paper {
```

```

bookTitleMarkup = \markup {
  \column {
    \fill-line { \fromproperty #'header:title }
    \null
    \wordwrap-field #'header:myText
  }
}

\markup {
  \null
}

```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\wordwrap` *args* (markup list)

Simple wordwrap. Use `\override #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

```

\markup {
  \wordwrap {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  }
}

```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (*#f*)
- `baseline-skip`

`\wordwrap-string` *arg* (string)

Wordwrap a string. Paragraphs may be separated with double newlines.

```

\markup {
  \override #'(line-width . 40)
  \wordwrap-string #"Lorem ipsum dolor sit amet, consectetur

```

adipisicing elit, sed do eiusmod tempor incididunt ut labore
et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat.

Excepteur sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum"

}

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore
et dolore magna aliqua.
Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo
consequat.
Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia
deserunt mollit anim id est laborum

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

A.10.3 Graphic

`\arrow-head` *axis* (integer) *dir* (direction) *filled* (boolean)

Produce an arrow head in specified direction and axis. Use the filled head if *filled* is specified.

```
\markup {
  \fontsize #5 {
    \general-align #Y #DOWN {
      \arrow-head #Y #UP ##t
      \arrow-head #Y #DOWN ##f
      \hspace #2
      \arrow-head #X #RIGHT ##f
      \arrow-head #X #LEFT ##f
    }
  }
}
```

▲ ▼ ➤ ➤

`\beam` *width* (number) *slope* (number) *thickness* (number)

Create a beam with the specified parameters.

```
\markup {
  \beam #5 #1 #2
}
```



`\bracket arg (markup)`
Draw vertical brackets around *arg*.

```
\markup {
  \bracket {
    \note #"2." #UP
  }
}
```



`\circle arg (markup)`
Draw a circle around *arg*. Use `thickness`, `circle-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \circle {
    Hi
  }
}
```



Used properties:

- `circle-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\draw-circle radius (number) thickness (number) filled (boolean)`
A circle of radius *radius* and thickness *thickness*, optionally filled.

```
\markup {
  \draw-circle #2 #0.5 ##f
  \hspace #2
  \draw-circle #2 #0 ##t
}
```



`\draw-hline`

Draws a line across a page, where the property `span-factor` controls what fraction of the page is taken up.

```
\markup {
  \column {
    \draw-hline
```

```

\override #'(span-factor . 1/3)
\draw-hline
}
}

```

Used properties:

- `span-factor` (1)
- `line-width`
- `draw-line-markup`

`\draw-line` *dest* (pair of numbers)

A simple line.

```

\markup {
  \draw-line #'(4 . 4)
  \override #'(thickness . 5)
  \draw-line #'(-3 . 0)
}

```



Used properties:

- `thickness` (1)

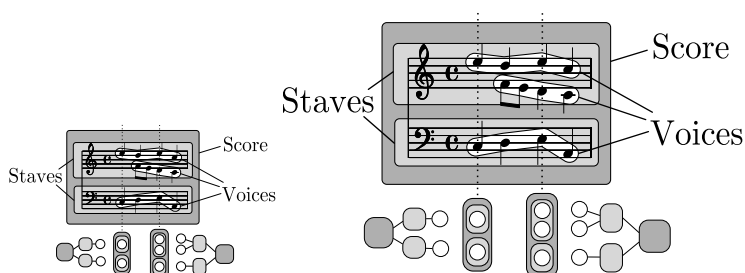
`\epsfile` *axis* (number) *size* (number) *file-name* (string)

Inline an EPS image. The image is scaled along *axis* to *size*.

```

\markup {
  \general-align #Y #DOWN {
    \epsfile #X #20 #"context-example.eps"
    \epsfile #Y #20 #"context-example.eps"
  }
}

```



`\filled-box` *xext* (pair of numbers) *yext* (pair of numbers) *blot* (number)

Draw a box with rounded corners of dimensions *xext* and *yext*. For example,

```
\filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0
```

creates a box extending horizontally from -0.3 to 1.8 and vertically from -0.3 up to 1.8, with corners formed from a circle of diameter 0 (i.e., sharp corners).

```

\markup {
  \filled-box #'(0 . 4) #'(0 . 4) #0
  \filled-box #'(0 . 2) #'(-4 . 2) #0.4
}

```

```

\filled-box #'(1 . 8) #'(0 . 7) #0.2
\with-color #white
\filled-box #'(-4.5 . -2.5) #'(3.5 . 5.5) #0.7
}

```



```

\hbracket arg (markup)
  Draw horizontal brackets around arg.
\markup {
  \hbracket {
    \line {
      one two three
    }
  }
}

```

one two three

```

\parenthesize arg (markup)
  Draw parentheses around arg. This is useful for parenthesizing a column containing
  several lines of text.

```

```

\markup {
  \line {
    \parenthesize {
      \column {
        foo
        bar
      }
    }
    \override #'(angularity . 2) {
      \parenthesize {
        \column {
          bah
          baz
        }
      }
    }
  }
}

```

$\left(\begin{smallmatrix} \text{foo} \\ \text{bar} \end{smallmatrix}\right) \left(\begin{smallmatrix} \text{bah} \\ \text{baz} \end{smallmatrix}\right)$

Used properties:

- width (0.25)
- thickness (1)
- size (1)

- `padding`
- `angularity` (0)

`\path` *thickness* (number) *commands* (list)

Draws a path with line thickness *thickness* according to the directions given in *commands*. *commands* is a list of lists where the `car` of each sublist is a drawing command and the `cdr` comprises the associated arguments for each command.

Line-cap styles and line-join styles may be customized by overriding the `line-cap-style` and `line-join-style` properties, respectively. Available line-cap styles are `'butt`, `'round`, and `'square`. Available line-join styles are `'miter`, `'round`, and `'bevel`.

The property `filled` specifies whether or not the path is filled with color.

There are seven commands available to use in the list `commands`: `moveto`, `rmoveto`, `lineto`, `rlineto`, `curveto`, `rcurveto`, and `closepath`. Note that the commands that begin with *r* are the relative variants of the other three commands.

The commands `moveto`, `rmoveto`, `lineto`, and `rlineto` take 2 arguments; they are the X and Y coordinates for the destination point.

The commands `curveto` and `rcurveto` create cubic Bézier curves, and take 6 arguments; the first two are the X and Y coordinates for the first control point, the second two are the X and Y coordinates for the second control point, and the last two are the X and Y coordinates for the destination point.

The `closepath` command takes zero arguments and closes the current subpath in the active path.

Note that a sequence of commands *must* begin with a `moveto` or `rmoveto` to work with the SVG output.

```
samplePath =
  #'((moveto 0 0)
    (lineto -1 1)
    (lineto 1 1)
    (lineto 1 -1)
    (curveto -5 -5 -5 5 -1 0)
    (closepath))
```

```
\markup {
  \path #0.25 #samplePath
}
```



Used properties:

- `filled` (`#f`)
- `line-join-style` (`round`)
- `line-cap-style` (`round`)

`\postscript` *str* (string)

This inserts *str* directly into the output as a PostScript command string.

```
ringsps = #
0.15 setlinewidth
0.9 0.6 moveto
0.4 0.6 0.5 0 361 arc
```

```

stroke
1.0 0.6 0.5 0 361 arc
stroke
"

rings = \markup {
  \with-dimensions #'(-0.2 . 1.6) #'(0 . 1.2)
  \postscript #ringsps
}

\relative c'' {
  c2^\rings
  a2_\rings
}

```



`\rounded-box` *arg* (markup)

Draw a box with rounded corners around *arg*. Looks at **thickness**, **box-padding** and **font-size** properties to determine line thickness and padding around the markup; the **corner-radius** property makes it possible to define another shape for the corners (default is 1).

```

c4^\markup {
  \rounded-box {
    Overtura
  }
}
c,8. c16 c4 r

```



Used properties:

- **box-padding** (0.5)
- **font-size** (0)
- **corner-radius** (1)
- **thickness** (1)

`\scale factor-pair` (pair of numbers) *arg* (markup)

Scale *arg*. *factor-pair* is a pair of numbers representing the scaling-factor in the X and Y axes. Negative values may be used to produce mirror images.

```

\markup {
  \line {
    \scale #'(2 . 1)
    stretched
    \scale #'(1 . -1)
    mirrored
  }
}

```


}

stretched willow`\triangle` *filled* (boolean)

A triangle, either filled or empty.

```

\markup {
  \triangle ##t
  \hspace #2
  \triangle ##f
}

```



Used properties:

- `baseline-skip` (2)
- `font-size` (0)
- `thickness` (0.1)

`\with-url` *url* (string) *arg* (markup)Add a link to URL *url* around *arg*. This only works in the PDF backend.

```

\markup {
  \with-url #"http://lilypond.org/" {
    LilyPond ... \italic {
      music notation for everyone
    }
  }
}

```

LilyPond ... *music notation for everyone*

A.10.4 Music

`\customTabClef` *num-strings* (integer) *staff-space* (number)

Draw a tab clef sans-serif style.

`\doubleflat`

Draw a double flat symbol.

```

\markup {
  \doubleflat
}

```

`\doublesharp`

Draw a double sharp symbol.

```

\markup {
  \doublesharp
}

```

`\flat`

Draw a flat symbol.

```
\markup {
  \flat
}
```



`\musicglyph glyph-name (string)`

glyph-name is converted to a musical symbol; for example, `\musicglyph #\"accidentals.natural\"` selects the natural sign from the music font. See [Section “The Feta font” dans *Manuel de notation*](#) for a complete listing of the possible glyphs.

```
\markup {
  \musicglyph #\"f\"
  \musicglyph #\"rests.2\"
  \musicglyph #\"clefs.G_change\"
}
```



`\natural`

Draw a natural symbol.

```
\markup {
  \natural
}
```



`\note-by-number log (number) dot-count (number) dir (number)`

Construct a note symbol, with stem. By using fractional values for *dir*, longer or shorter stems can be obtained.

```
\markup {
  \note-by-number #3 #0 #DOWN
  \hspace #2
  \note-by-number #1 #2 #0.8
}
```



Used properties:

- `style '()`
- `font-size (0)`

`\note duration (string) dir (number)`

This produces a note with a stem pointing in *dir* direction, with the *duration* for the note head type and augmentation dots. For example, `\note #\"4.\" #-0.75` creates a dotted quarter note, with a shortened down stem.

```
\markup {
  \override #'(style . cross) {
```

```

\note #"4.." #UP
}
\hspace #2
\note #"breve" #0
}

```

↓ .. 

Used properties:

- `style '()`
- `font-size (0)`

`\score score (score)`

Inline an image of music.

```

\markup {
  \score {
    \new PianoStaff <<
      \new Staff \relative c' {
        \key f \major
        \time 3/4
        \mark \markup { Allegro }
        f2\p( a4)
        c2( a4)
        bes2( g'4)
        f8( e) e4 r
      }
      \new Staff \relative c {
        \clef bass
        \key f \major
        \time 3/4
        f8( a c a c a
        f c' es c es c)
        f,( bes d bes d bes)
        f( g bes g bes g)
      }
    >>
  }
  \layout {
    indent = 0.0\cm
    \context {
      \Score
      \override RehearsalMark
        #'break-align-symbols = #'(time-signature key-signature)
      \override RehearsalMark
        #'self-alignment-X = #LEFT
    }
    \context {
      \Staff
      \override TimeSignature
        #'break-align-anchor-alignment = #LEFT
    }
  }
}

```

}



Used properties:

- `baseline-skip`

`\semiflat`

Draw a semiflat symbol.

```
\markup {
  \semiflat
}
```

`\semisharp`

Draw a semisharp symbol.

```
\markup {
  \semisharp
}
```

`\sesquiflat`

Draw a 3/2 flat symbol.

```
\markup {
  \sesquiflat
}
```

`\sesquisharp`

Draw a 3/2 sharp symbol.

```
\markup {
  \sesquisharp
}
```

`\sharp`

Draw a sharp symbol.

```
\markup {
  \sharp
}
```

#

`\tied-lyric` *str* (string)

Like simple-markup, but use tie characters for ‘~’ tilde symbols.

```
\markup \column {
  \tied-lyric #"Siam navi~all'onde~algenti Lasciate~in abbandono"
  \tied-lyric #"Impetuousi venti I nostri~affetti sono"
  \tied-lyric #"Ogni diletto~e scoglio Tutta la vita~e~un mar."
}
```

Siam naviall'onde algenti Lasciatein abbandono
 Impetuousi venti I nostriaffetti sono
 Ogni dilettoe scoglio Tutta la vitae un mar.

Used properties:

- `word-space`

A.10.5 Instrument Specific Markup

`\fret-diagram` *definition-string* (string)

Make a (guitar) fret diagram. For example, say

```
\markup \fret-diagram #"s:0.75;6-x;5-x;4-o;3-2;2-3;1-2;"
```

for fret spacing 3/4 of staff space, D chord diagram

Syntax rules for *definition-string*:

- Diagram items are separated by semicolons.
- Possible items:
 - `s: number` – Set the fret spacing of the diagram (in staff spaces). Default: 1.
 - `t: number` – Set the line thickness (relative to normal line thickness). Default: 0.5.
 - `h: number` – Set the height of the diagram in frets. Default: 4.
 - `w: number` – Set the width of the diagram in strings. Default: 6.
 - `f: number` – Set fingering label type (0 = none, 1 = in circle on string, 2 = below string). Default: 0.
 - `d: number` – Set radius of dot, in terms of fret spacing. Default: 0.25.
 - `p: number` – Set the position of the dot in the fret space. 0.5 is centered; 1 is on lower fret bar, 0 is on upper fret bar. Default: 0.6.
 - `c: string1-string2-fret` – Include a barre mark from *string1* to *string2* on *fret*.
 - `string-fret` – Place a dot on *string* at *fret*. If *fret* is ‘o’, *string* is identified as open. If *fret* is ‘x’, *string* is identified as muted.
 - `string-fret-fingering` – Place a dot on *string* at *fret*, and label with *fingering* as defined by the `f:` code.
- Note: There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)

- `align-dir` (-0.4)

`\fret-diagram-terse` *definition-string* (string)

Make a fret diagram markup using terse string-based syntax.

Here is an example

```
\markup \fret-diagram-terse #"x;x;o;2;3;2;"
```

for a D chord diagram.

Syntax rules for *definition-string*:

- Strings are terminated by semicolons; the number of semicolons is the number of strings in the diagram.
- Mute strings are indicated by ‘x’.
- Open strings are indicated by ‘o’.
- A number indicates a fret indication at that fret.
- If there are multiple fret indicators desired on a string, they should be separated by spaces.
- Fingerings are given by following the fret number with a -, followed by the finger indicator, e.g. ‘3-2’ for playing the third fret with the second finger.
- Where a barre indicator is desired, follow the fret (or fingering) symbol with -(to start a barre and -) to end the barre.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-verbose` *marking-list* (pair)

Make a fret diagram containing the symbols indicated in *marking-list*.

For example,

```
\markup \fret-diagram-verbose
#'( (mute 6) (mute 5) (open 4)
    (place-fret 3 2) (place-fret 2 3) (place-fret 1 2) )
```

produces a standard D chord diagram without fingering indications.

Possible elements in *marking-list*:

`(mute string-number)`

Place a small ‘x’ at the top of string *string-number*.

`(open string-number)`

Place a small ‘o’ at the top of string *string-number*.

`(barre start-string end-string fret-number)`

Place a barre indicator (much like a tie) from string *start-string* to string *end-string* at fret *fret-number*.

`(capo fret-number)`

Place a capo indicator (a large solid bar) across the entire fretboard at fret location *fret-number*. Also, set fret *fret-number* to be the lowest fret on the fret diagram.

`(place-fret string-number fret-number [finger-value [color-modifier]])`

Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*, and an optional

color modifier *color-modifier*. By default, the fret playing indicator is a solid dot. This can be globally changed by setting the value of the variable *dot-color*. Setting *color-modifier* to **inverted** inverts the dot color for a specific fingering. If the *finger* part of the **place-fret** element is present, *finger-value* will be displayed according to the setting of the variable *finger-code*. There is no limit to the number of fret indications per string.

Used properties:

- **thickness** (0.5)
- **fret-diagram-details**
- **size** (1.0)
- **align-dir** (-0.4)

\harp-pedal *definition-string* (string)

Make a harp pedal diagram.

Possible elements in *definition-string*:

- ^ pedal is up
- pedal is neutral
- v pedal is down
- | vertical divider line
- o the following pedal should be circled (indicating a change)

The function also checks if the string has the typical form of three pedals, then the divider and then the remaining four pedals. If not it prints out a warning. However, in any case, it will also print each symbol in the order as given. This means you can place the divider (even multiple dividers) anywhere you want, but you'll have to live with the warnings.

The appearance of the diagram can be tweaked *inter alia* using the *size* property of the *TextScript* grob (**\override Voice.TextScript #'size = #0.3**) for the overall, the *thickness* property (**\override Voice.TextScript #'thickness = #3**) for the line thickness of the horizontal line and the divider. The remaining configuration (box sizes, offsets and spaces) is done by the *harp-pedal-details* list of properties (**\override Voice.TextScript #'harp-pedal-details #'box-width = #1**). It contains the following settings: **box-offset** (vertical shift of the box center for up/down pedals), **box-width**, **box-height**, **space-before-divider** (the spacing between two boxes before the divider) and **space-after-divider** (box spacing after the divider).

\markup \harp-pedal #"^-v|--ov^"



Used properties:

- **thickness** (0.5)
- **harp-pedal-details** ('')
- **size** (1.2)

\woodwind-diagram *instrument* (symbol) *user-draw-commands* (list)

Make a woodwind-instrument diagram. For example, say

```
\markup \woodwind-diagram
  #'oboe #'((lh . (d ees)) (cc . (five3qT1q)) (rh . (gis)))
```

for an oboe with the left-hand d key, left-hand ees key, and right-hand gis key depressed while the five-hole of the central column effectuates a trill between 1/4 and 3/4 closed.

The following instruments are supported:

- piccolo
- flute
- oboe
- clarinet
- bass-clarinet
- saxophone
- bassoon
- contrabassoon

To see all of the callable keys for a given instrument, include the function (**print-keys 'instrument**) in your .ly file, where instrument is the instrument whose keys you want to print.

Certain keys allow for special configurations. The entire gamut of configurations possible is as follows:

- 1q (1/4 covered)
- 1h (1/2 covered)
- 3q (3/4 covered)
- R (ring depressed)
- F (fully covered; the default if no state put)

Additionally, these configurations can be used in trills. So, for example, **three3qTR** effectuates a trill between 3/4 full and ring depressed on the three hole. As another example, **threeRT** effectuates a trill between R and open, whereas **threeTR** effectuates a trill between open and shut. To see all of the possibilities for all of the keys of a given instrument, invoke (**print-keys-verbose 'instrument**).

Lastly, substituting an empty list for the pressed-key alist will result in a diagram with all of the keys drawn but none filled, for example:

```
\markup \woodwind-diagram #'oboe #'()
```

Used properties:

- **graphical** (#t)
- **thickness** (0.1)
- **size** (1)

A.10.6 Other

```
\auto-footnote mkup (markup) note (markup)
```

Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \auto-footnote a b
  \override #'(padding . 0.2)
  \auto-footnote c d
}
```


a c

The footnote will be annotated automatically.

Used properties:

- padding (0.0)
- raise (0.5)

`\backslashed-digit` *num* (integer)

A feta number, with backslash. This is for use in the context of figured bass notation.

```
\markup {
  \backslashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \backslashed-digit #7
}
```

5 7

Used properties:

- thickness (1.6)
- font-size (0)

`\char` *num* (integer)

Produce a single character. Characters encoded in hexadecimal format require the prefix `#x`.

```
\markup {
  \char #65 \char ##x00a9
}
```

A ©

`\eyeglasses`

Prints out eyeglasses, indicating strongly to look at the conductor.

```
\markup { \eyeglasses }
```

60'

`\footnote` *mkup* (markup) *note* (markup)

Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \auto-footnote a b
  \override #'(padding . 0.2)
  \auto-footnote c d
}
```

a c

The footnote will not be annotated automatically.

`\fraction` *arg1* (markup) *arg2* (markup)

Make a fraction of two markups.

```
\markup {
  \fraction 355 113
}
```

$$\pi \approx \frac{355}{113}$$

Used properties:

- `font-size (0)`

`\fromproperty symbol (symbol)`

Read the *symbol* from property settings, and produce a stencil from the markup contained within. If *symbol* is not defined, it returns an empty markup.

```
\header {
  myTitle = "myTitle"
  title = \markup {
    from
    \italic
    \fromproperty #'header:myTitle
  }
}
\markup {
  \null
}
```

from *myTitle*

`\left-brace size (number)`

A feta brace in point size *size*.

```
\markup {
  \left-brace #35
  \hspace #2
  \left-brace #45
}
```

{ }

`\lookup glyph-name (string)`

Lookup a glyph by name.

```
\markup {
  \override #'(font-encoding . fetaBraces) {
    \lookup #"brace200"
    \hspace #2
    \rotate #180
    \lookup #"brace180"
  }
}
```

}

{ }

`\markalphabet` *num* (integer)

Make a markup letter for *num*. The letters start with A to Z and continue with double letters.

```
\markup {
  \markalphabet #8
  \hspace #2
  \markalphabet #26
}
```

I AA

`\markletter` *num* (integer)

Make a markup letter for *num*. The letters start with A to Z (skipping letter I), and continue with double letters.

```
\markup {
  \markletter #8
  \hspace #2
  \markletter #26
}
```

J AB

`\null`

An empty markup with extents of a single point.

```
\markup {
  \null
}
```

`\on-the-fly` *procedure* (procedure) *arg* (markup)

Apply the *procedure* markup command to *arg*. *procedure* should take a single argument.

`\override` *new-prop* (pair) *arg* (markup)

Add the argument *new-prop* to the property list. Properties may be any property supported by Section “font-interface” dans *Référence des propriétés internes*, Section “text-interface” dans *Référence des propriétés internes* and Section “instrument-specific-markup-interface” dans *Référence des propriétés internes*.

```
\markup {
  \line {
    \column {
      default
    }
  }
}
```

```

        baseline-skip
    }
    \hspace #2
    \override #'(baseline-skip . 4) {
        \column {
            increased
            baseline-skip
        }
    }
}
}

```

default	increased
baseline-skip	baseline-skip

`\page-link` *page-number* (number) *arg* (markup)

Add a link to the page *page-number* around *arg*. This only works in the PDF backend.

```

\markup {
  \page-link #2 { \italic { This links to page 2... } }
}

```

This links to page 2...

`\page-ref` *label* (symbol) *gauge* (markup) *default* (markup)

Reference to a page number. *label* is the label set on the referenced page (using the `\label` command), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

`\pattern` *count* (integer) *axis* (integer) *space* (number) *pattern* (markup)

Prints *count* times a *pattern* markup. Patterns are spaced apart by *space*. Patterns are distributed on *axis*.

```

\markup \column {
  "Horizontally repeated : "
  \pattern #7 #X #2 \flat
  \null
  "Vertically repeated : "
  \pattern #3 #Y #0.5 \flat
}

```

Horizontally repeated :

b b b b b b b

Vertically repeated :

b
b
b

`\property-recursive` *symbol* (symbol)

Print out a warning when a header field markup contains some recursive markup definition.

`\right-brace` *size* (number)

A feta brace in point size *size*, rotated 180 degrees.

```
\markup {
  \right-brace #45
  \hspace #2
  \right-brace #35
}
```

{ }

`\slashed-digit` *num* (integer)

A feta number, with slash. This is for use in the context of figured bass notation.

```
\markup {
  \slashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \slashed-digit #7
}
```

5 7

Used properties:

- `thickness` (1.6)
- `font-size` (0)

`\stencil` *stil* (stencil)

Use a stencil as markup.

```
\markup {
  \stencil #(make-circle-stencil 2 0 #t)
}
```



`\strut`

Create a box of the same height as the space in the current font.

`\transparent` *arg* (markup)

Make *arg* transparent.

```
\markup {
  \transparent {
    invisible text
  }
}
```

`\verbatim-file` *name* (string)

Read the contents of file *name*, and include it verbatim.

```

\markup {
  \verbatim-file #"simple.ly"
}

%% A simple piece in LilyPond, a scale.
\relative c' {
  c d e f g a b c
}
%% Optional helper for automatic updating by convert-ly.
%% May be omitted.
\version "2.16.0"

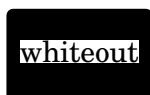
```

`\whiteout` *arg* (markup)
Provide a white background for *arg*.

```

\markup {
  \combine
    \filled-box #'(-1 . 10) #'(-3 . 4) #1
    \whiteout whiteout
}

```



`\with-color` *color* (color) *arg* (markup)
Draw *arg* in color specified by *color*.

```

\markup {
  \with-color #red
  red
  \hspace #2
  \with-color #green
  green
  \hspace #2
  \with-color #blue
  blue
}

```

red green blue

`\with-dimensions` *x* (pair of numbers) *y* (pair of numbers) *arg* (markup)
Set the dimensions of *arg* to *x* and *y*.

`\with-link` *label* (symbol) *arg* (markup)
Add a link to the page holding label *label* around *arg*. This only works in the PDF backend.

```

\markup {
  \with-link #'label {
    \italic { This links to the page containing the label... }
  }
}

```

This links to the page containing the label...

A.11 Commandes pour liste de markups

Les commandes suivantes peuvent être utilisées dans un bloc `\markuplist { }`.

`\column-lines` *args* (markup list)

Like `\column`, but return a list of lines instead of a single markup. `baseline-skip` determines the space between each markup in *args*.

Used properties:

- `baseline-skip`

`\justified-lines` *args* (markup list)

Like `\justify`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (*#f*)
- `baseline-skip`

`\override-lines` *new-prop* (pair) *args* (markup list)

Like `\override`, for markup lists.

`\table-of-contents`

`\wordwrap-internal` *justify* (boolean) *args* (markup list)

Internal markup list command used to define `\justify` and `\wordwrap`.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (*#f*)

`\wordwrap-lines` *args* (markup list)

Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (*#f*)
- `baseline-skip`

`\wordwrap-string-internal` *justify* (boolean) *arg* (string)

Internal markup list command used to define `\justify-string` and `\wordwrap-string`.

Used properties:



































































- `text-direction` (1)
- `word-space`
- `line-width`

A.12 Liste des caractères spéciaux

Voici une table des caractères spéciaux disponibles. Pour plus de précisions, voir [\[Équivalents ASCII\]](#), page 482.

Cette liste utilise la syntaxe HTML, à l'instar de la plupart des caractères qui la composent ; les autres sont inspirés du langage L^AT_EX.

Les caractères sont ici inclus dans une boîte, de façon à mettre leur taille en évidence, et un léger décalage a été appliqué pour les décoller de l'encadrement.

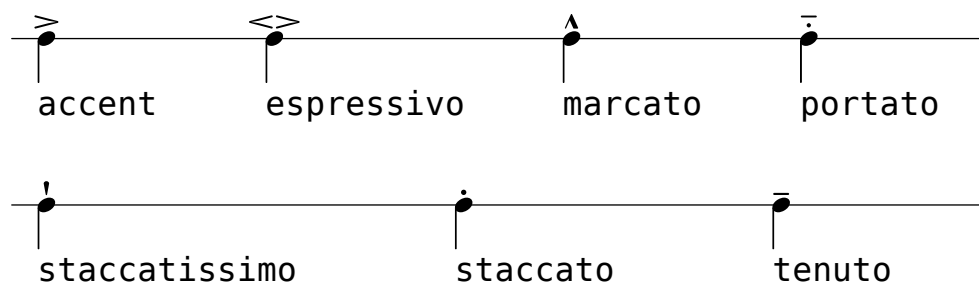
<code>&hellip;</code>		<code>&ndash;</code>		<code>&mdash;</code>		<code>&iexcl;</code>	
<code>&iquest;</code>		<code>&solidus;</code>		<code>&flq;</code>		<code>&frq;</code>	
<code>&flqq;</code>		<code>&frqq;</code>		<code>&glq;</code>		<code>&grq;</code>	
<code>&glqq;</code>		<code>&grqq;</code>		<code>&elq;</code>		<code>&erq;</code>	
<code>&elqq;</code>		<code>&erqq;</code>		<code>&ensp;</code>		<code>&emsp;</code>	
<code>&thinsp;</code>		<code>&nbsp;</code>		<code>&nnbsp;</code>		<code>&zwj;</code>	
<code>&zwj;</code>		<code>&middot;</code>		<code>&bull;</code>		<code>&copyright;</code>	
<code>&registered;</code>		<code>&trademark;</code>		<code>&dagger;</code>		<code>&Dagger;</code>	
<code>&numero;</code>		<code>&ordf;</code>		<code>&ordm;</code>		<code>&para;</code>	
<code>&sect;</code>		<code>&deg;</code>		<code>&numero;</code>		<code>&permil;</code>	
<code>&brvbar;</code>		<code>&acute;</code>		<code>&acutedbl;</code>		<code>&grave;</code>	
<code>&breve;</code>		<code>&caron;</code>		<code>&cedilla;</code>		<code>&circumflex;</code>	
<code>&diaeresis;</code>		<code>&macron;</code>		<code>&aa;</code>		<code>&AA;</code>	
<code>&ae;</code>		<code>&AE;</code>		<code>&dh;</code>		<code>&DH;</code>	
<code>&dj;</code>		<code>&DJ;</code>		<code>&l;</code>		<code>&L;</code>	
<code>&ng;</code>		<code>&NG;</code>		<code>&o;</code>		<code>&O;</code>	
<code>&oe;</code>		<code>&OE;</code>		<code>&s;</code>		<code>&ss;</code>	

$\&th;$	\boxed{p} $\&TH;$	\boxed{P} $\+$	$\boxed{+}$ $\−$	\equiv
$\×$	$\boxed{\times}$ $\÷$	$\boxed{\div}$ $\¹$	$\boxed{1}$ $\²$	$\boxed{2}$
$\³$	$\boxed{3}$ $\&sqr;$	$\boxed{\sqrt{}}$ $\&increment;$	$\boxed{\Delta}$ $\&infty;$	∞
$\∑$	$\boxed{\Sigma}$ $\±$	$\boxed{\pm}$ $\•$	$\boxed{\partial}$ $\&partial;$	∂
$\&neg;$	$\boxed{\pounds}$ $\¤cy;$	$\boxed{\$}$ $\$$	$\boxed{\text{€}}$ $\€$	€
$\£s;$	£ $\¥$	¥ $\¢$	¢	

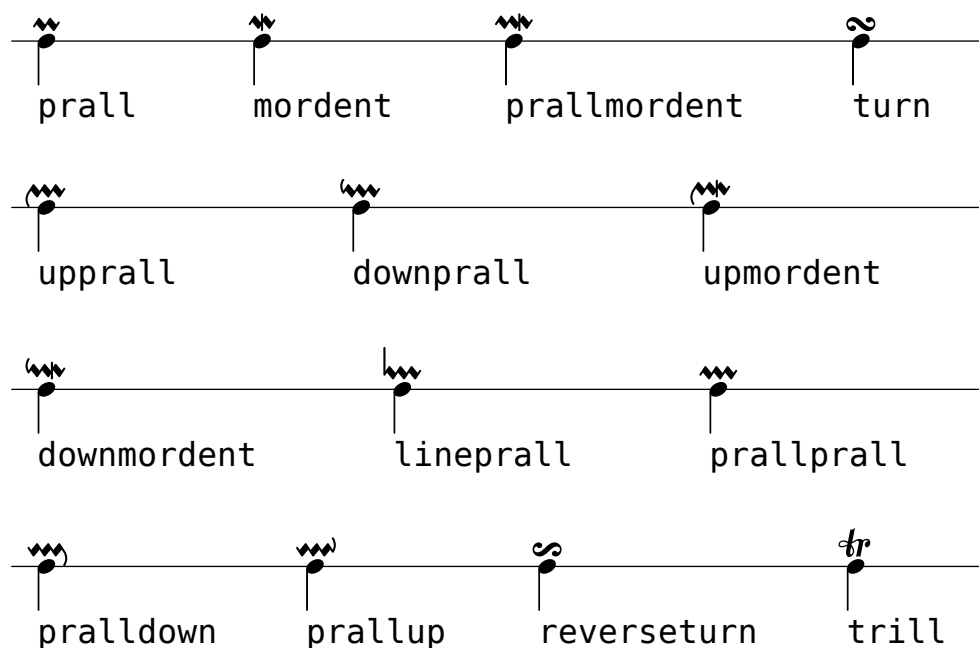
A.13 Liste des signes d'articulation

Voici la liste exhaustive des symboles prédéfinis et que vous pouvez attacher à une note (par ex. `c\accent`) :

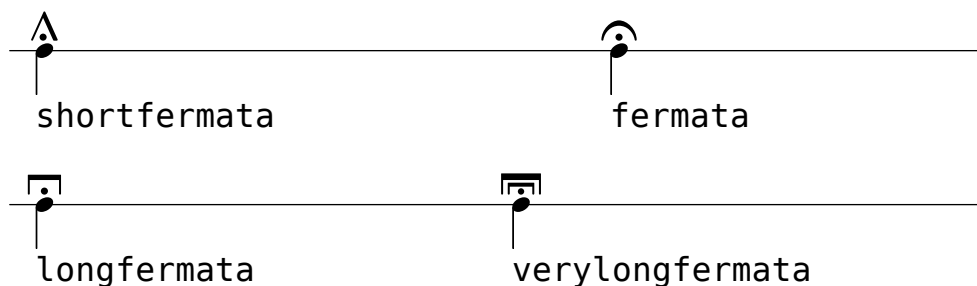
Scripts d'articulation



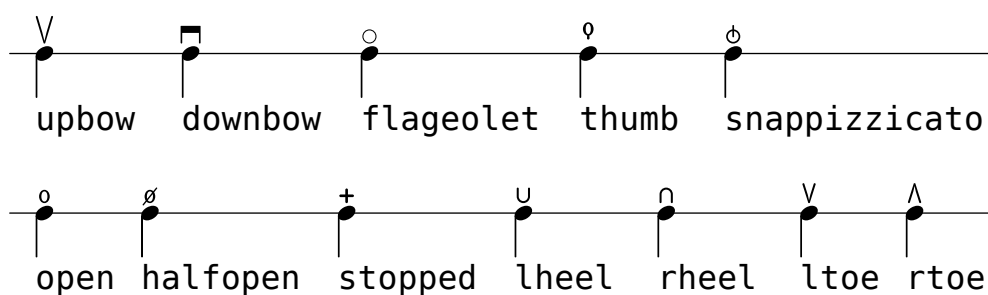
Scripts d'ornement



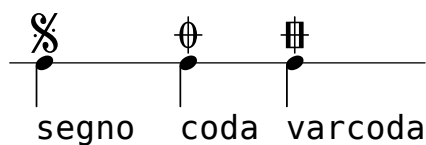
Scripts de point d'orgue et point d'arrêt



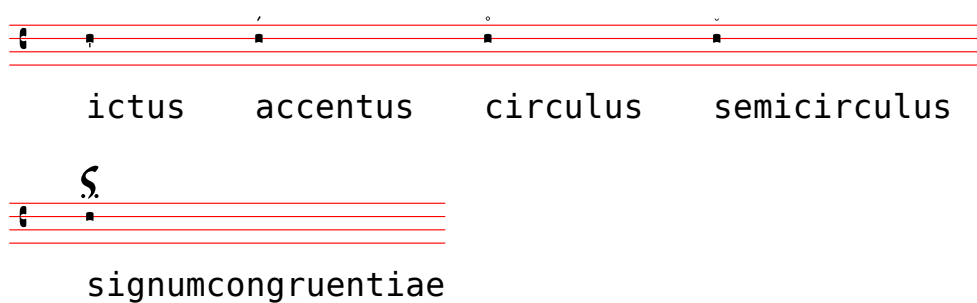
Scripts spécifiques à certains instruments



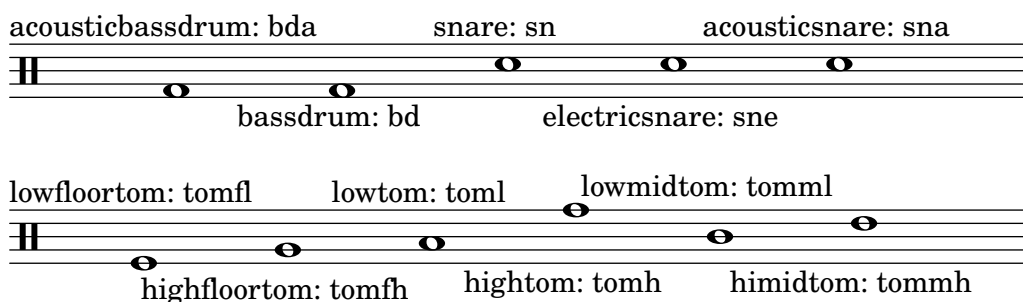
Scripts de reprise et de répétition



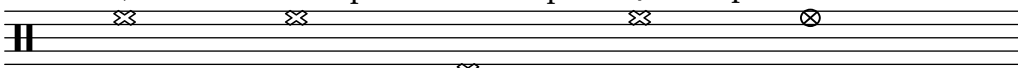
Scripts pour musique ancienne



A.14 Notes utilisées en percussion

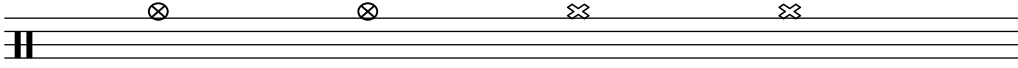


closedhihat: hhc pedalhihat: hhp halfopenhihat: hhho



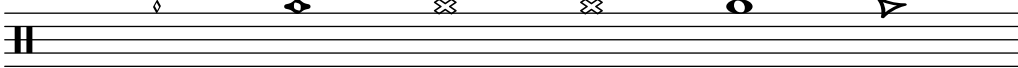
hihat: hh openhihat: hho

crashcymbala: cymca ridecymbala: cymra



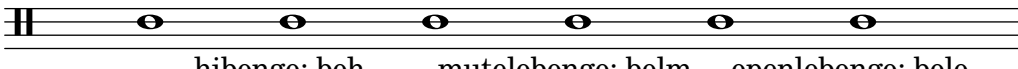
crashcymbal: cymc ridecymbal: cymr

chinesecymbal: cymch crashcymbalb: cymcb ridebell: rb



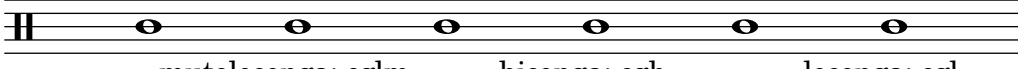
splashcymbal: cymc ridecymbalb: cymrb cowbell: cb

mutehibongo: boh openhibongo: boho lobongo: bol



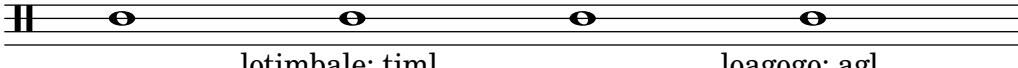
hibongo: boh mutelobongo: bolm openlobongo: bolo

mutehiconga: cghm openhiconga: cghe openloconga: cglo



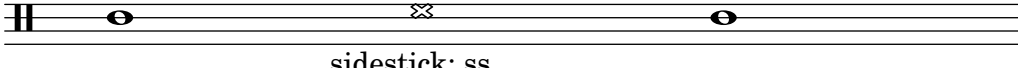
muteloconga: cglm hiconga: cgh loconga: cgl

hitimbale: timh hiagogo: agh




lotimbale: timl loagogo: agl

hisidestick: ssh losidestick: ssl



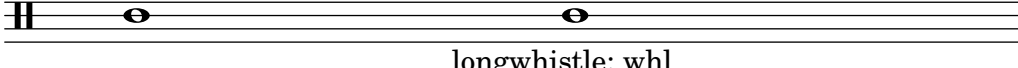
sidestick: ss

shortguiro: guis guiro: gui maracas: mar



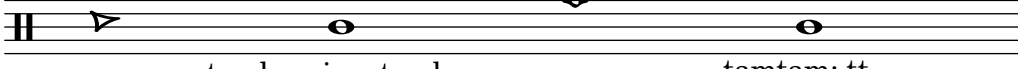
longguiro: guil cabasa: cab

shortwhistle: whs



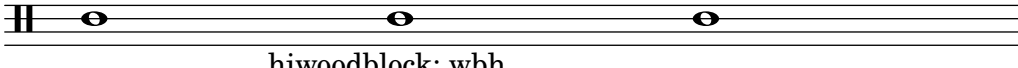
longwhistle: whl

handclap: hc vibraslap: vib


















tambourine: tamb tamtam: tt

claves: cl lowoodblock: wbl



hiwoodblock: wbh

mutecuica: cuim	mutetriangle: trim	opentriangle: trio
		
opencuica: cuio	triangle: tri	
		
oneup: ua	threeup: uc	fiveup: ue
		
twoup: ub	fourup: ud	
		
onedown: da	threedown: dc	fivedown: de
		
twodown: db	fourdown: dd	
		

A.15 Glossaire technique

Ce glossaire regroupe les termes techniques et concepts utilisés en interne par LilyPond. Ils apparaissent aussi bien dans les manuels, que sur les listes de diffusion et dans le code source.

alist (liste associative)

Une liste associative – **alist** pour *association list* – est une paire Scheme qui associe une valeur à une clé : (clé . valeur). Le fichier ‘scm/lily.scm’ contient par exemple une *alist* « type-p-name-alist » qui associe certains types de prédicat (p. ex. ly:music?) à des noms (p. ex. « music ») de telle sorte qu’une erreur lors d’un contrôle de typage puisse être rapportée en console avec mention du type de prédicat attendu.

callback (rappel)

Un **rappel**, *callback* en anglais, est une routine, fonction ou méthode qui est passée en argument à une autre fonction. Cette dernière peut alors faire usage de cette fonction de rappel comme de n’importe quelle autre fonction, alors qu’elle ne la connaît pas par avance. Cette façon de procéder permet à des couches logicielles de bas niveau d’appeler des fonctions définies à des niveaux plus élevés. LilyPond utilise abondamment les *callbacks* afin que le code Scheme saisi par l’utilisateur puisse définir les actions de bas niveau à opérer.

closure (clôture)

En Scheme, une **clôture** (en anglais, *closure*) est créée lorsqu’une fonction, généralement une expression lambda, est passée en tant que variable. La clôture comporte, en plus du code de la fonction, des références à des variables libres dans l’environnement lexical – variables utilisées dans l’expression mais définies ailleurs. Lorsque la fonction est par la suite appliquée aux différents arguments, les références aux variables libres, capturées par la clôture, permettent d’obtenir la valeur de ces variables libres qui sera utilisée lors des calculs. L’une des propriétés intéressantes des clôtures est la rétention de la valeur de variables internes tout au long des différentes invocations, leur état étant alors persistant.

Une **clôture simple** – *simple closure* en anglais – est une clôture ne comportant pas de variable libre, et donc pas de référence à des variables libres.

Une clôture simple est représentée dans LilyPond par un objet Scheme (un *smob*) contenant l’expression et la méthode pour appliquer cette expression à la liste d’arguments qui lui est passée.

glyphe

Un **glyphe** est une représentation graphique particulière d’un caractère typographique ou d’une combinaison de deux caractères formant une ligature. Un jeu de glyphes aux mêmes style et allure constitue une fonte ; un jeu de fontes comprenant plusieurs styles et tailles constitue un type de caractère.

Voir aussi

Manuel de notation : [Section 1.8.3 \[Fontes\]](#), page 234, [Section 3.3.3 \[Caractères spéciaux\]](#), page 480.

grob (objet graphique)

Dans LilyPond, les objets représentant les différents éléments de notation, comme les têtes de note, hampes, liaisons, doigtés, clefs etc. sont appelés « objets de rendu » ou objet graphique – en anglais *G*Raphical *O*Bjects couramment abrégé en **grob**. Ils sont représentés par des instances de la classe **grob**.

Voir aussi

Manuel d’initiation : [Section “Objets et interfaces”](#) dans *Manuel d’initiation*, [Section “Conventions de nommage des objets et propriétés”](#) dans *Manuel d’initiation*, [Section “Propriétés des objets de rendu”](#) dans *Manuel d’initiation*.

Référence des propriétés internes : [Section “grob-interface”](#) dans *Référence des propriétés internes*, [Section “All layout objects”](#) dans *Référence des propriétés internes*.

inaltérable

Un objet est dit **inaltérable** – *immutable* en anglais – dès lors que son état ne peut être modifié après sa création ; les objets altérables, à l’inverse, peuvent être modifiés après création.

Pour LilyPond, une propriété est inaltérable ou commune lorsqu’elle définit le style ou le comportement par défaut d’objets graphiques ; une telle propriété est partagée par un certain nombre de *grobs*. En apparence contradiction avec ce que leur nom laisse accroire, de telles propriétés peuvent être adaptées par `\override` et `\revert`.

Voir aussi

Manuel de notation : [\[altérable\]](#), page 692.

interface

Les actions et propriétés communes à plusieurs objets graphiques sont regroupées dans un objet appelé **grob-interface**, ou « interface » pour faire court.

Voir aussi

Manuel d’initiation : [Section “Objets et interfaces”](#) dans *Manuel d’initiation*, [Section “Conventions de nommage des objets et propriétés”](#) dans *Manuel d’initiation*, [Section “Propriétés listées par interface”](#) dans *Manuel d’initiation*.

Manuel de notation : [Section 5.2.2 \[Interfaces de rendu\]](#), page 566.

Référence des propriétés internes : [Section “Graphical Object Interfaces”](#) dans *Référence des propriétés internes*.

lexer (analyseur lexical)

Un **lexer** est un programme chargé de convertir une séquence de caractères en une séquence de jetons. Cette opération s’appelle l’analyse lexicale. L’analyseur lexical de LilyPond convertit le flot d’information contenu dans un fichier ‘.ly’ en flot de jetons qui pourront être traités lors de l’étape suivante, l’analyse grammaticale abordée à la rubrique [\[parser \(analyseur syntaxique\)\]](#), page 692. L’analyseur lexical de LilyPond repose sur **Flex** ; les règles lexicales sont regroupées dans le fichier ‘lily/lexer.ll’. Ce fichier, partie intégrante des sources, n’est pas distribué avec les programmes binaires de LilyPond.

altérable

Un objet est dit **altérable** – *mutable* en anglais – lorsque son état est sujet à modification après sa création, à l'inverse des objets inaltérables dont l'état est figé dès leur création.

Les propriétés altérables contiennent, pour LilyPond, des valeurs spécifiques à un objet graphique. En particulier, les listes d'autres objets ou résultats de calculs sont enregistrés sous forme de propriétés altérables.

Voir aussi

Manuel de notation : [\[inaltérable\]](#), page 691.

output-def (définition de sortie)

Une instance de la classe `Output-def` contient les méthodes et structures des données associées à un bloc de sortie. Ces instances sont créées par les blocs `\midi`, `\layout` et `\paper`.

parser (analyseur syntaxique)

Un analyseur syntaxique – **parser** en anglais – est un programme qui analyse la séquence de jetons produite par l'analyseur lexical pour en déterminer la structure grammaticale. Les jetons sont, pour ce faire, regroupés progressivement en tronçons plus importants, selon des règles grammaticales. Lorsque la séquence de jetons est valide, le produit final est une arborescence de jetons ayant à sa base le symbole grammatical de début. Dès lors que cette étape n'est pas concluante, le fichier est déclaré invalide ; un message approprié est alors émis. Les différents regroupements syntaxiques ainsi que les règles de construction des regroupements relatifs à la grammaire de LilyPond sont définis dans le fichier '`lily/parser.yy`' et présentés selon la forme de Backus-Naur (BNF) à l'annexe [Annexe C \[Grammaire de LilyPond\]](#), page 758. Ce fichier est utilisé par le générateur de *parser* Bison lors de la construction du programme. Partie intégrante des sources, il n'est pas distribué avec les programmes binaires de LilyPond.

variable de l'analyseur grammatical

Il s'agit de variables définies directement en Scheme. Dans la mesure où leur champ sémantique peut porter à confusion, il est fortement déconseillé de les utiliser tels quels.

La modification des valeurs de l'une de ces variables dans un fichier '`.ly`' sera effective de manière globale. Les valeurs modifiées, sauf à être explicitement remises à leur état d'origine, affecteront tous les blocs `\score` rencontrés, y compris s'ils proviennent d'autres fichiers ajoutés par une commande `\include`. Ceci peut avoir des conséquences inattendues et les erreurs qui pourraient en découler difficiles à localiser dans le cadre d'un projet d'envergure.

LilyPond utilise les variables suivantes :

- `afterGraceFraction`
- `musicQuotes`
- `mode`
- `output-count`
- `output-suffix`
- `parseStringResult`
- `partCombineListener`
- `pitchnames`
- `toplevel-bookparts`
- `toplevel-scores`
- `showLastLength`
- `showFirstLength`

prob (objet de propriété)

Les objets de propriété – **probs** pour *Property Objects* – sont des instances de la classe **Prob**, une classe de base simple pour les objets qui disposent de de listes associatives de propriétés altérables et inaltérables ainsi que les méthodes pour les manipuler. Les classes **Music** et **Stream_event** dérivent d'un **prob**. Les instances de la classe **prob** se créent aussi pour garder trace du contenu des systèmes une fois formatés et des blocs de titrage lors de la phase de mise en forme des pages.

clôture simple

Voir [\[closure \(clôture\)\]](#), page 690.

smob (objet Scheme)

Les objets Scheme – **Smobs** pour *Scheme Objects* – font partie du mécanisme utilisé par l'interpréteur Guile pour exporter en code Scheme les objets C ou C++. Dans LilyPond, les *smobs* sont créés, grâce à des macros, à partir d'objets C++. On peut distinguer deux types d'objets *smob* : des *smobs* simples destinés aux objets inaltérables comme les nombres par exemples, et des *smobs* complexes utilisés pour des objets possédant une identité. De plus amples informations sont disponibles dans les sources de LilyPond, au sein du fichier 'lily/includes/smob.hh'.

stencil

Une instance de la classe **stencil** comporte l'information nécessaire à l'impression d'un objet typographique. Il s'agit d'un *smob* simple qui contient un espace de confinement qui définit l'envergure verticale et horizontale de l'objet ainsi qu'une expression Scheme qui imprimera l'objet après évaluation. Les stencils peuvent se combiner et adopter une forme plus complexe définie par une arborescence d'expressions Scheme des stencils qui la composent.

La propriété **stencil**, qui permet de connecter un *grob* à son stencil, est définie par l'interface **grob-interface**.

Voir aussi

Référence des propriétés internes : [Section “grob-interface” dans Référence des propriétés internes](#).

A.16 Liste des propriétés de contexte

additionalPitchPrefix (string)

Text with which to prefix additional pitches within a chord name.

aDueText (markup)

Text to print at a unisono passage.

alignAboveContext (string)

Where to insert newly created context in vertical alignment.

alignBassFigureAccidentals (boolean)

If true, then the accidentals are aligned in bass figure context.

alignBelowContext (string)

Where to insert newly created context in vertical alignment.

alternativeNumberingStyle (symbol)

The style of an alternative's bar numbers. Can be **numbers** for going back to the same number or **numbers-with-letters** for going back to the same number with letter suffixes. No setting will not go back in measure-number time.

associatedVoice (string)

Name of the **Voice** that has the melody for this **Lyrics** line.

autoAccidentals (list)

List of different ways to typeset an accidental.

For determining when to print an accidental, several different rules are tried. The rule that gives the highest number of accidentals is used.

Each entry in the list is either a symbol or a procedure.

symbol The symbol is the name of the context in which the following rules are to be applied. For example, if *context* is **Section “Score” dans Référence des propriétés internes** then all staves share accidentals, and if *context* is **Section “Staff” dans Référence des propriétés internes** then all voices in the same staff share accidentals, but staves do not.

procedure The procedure represents an accidental rule to be applied to the previously specified context.

The procedure takes the following arguments:

context The current context to which the rule should be applied.

pitch The pitch of the note to be evaluated.

barnum The current bar number.

measurepos

The current measure position.

The procedure returns a pair of booleans. The first states whether an extra natural should be added. The second states whether an accidental should be printed. (**#t** . **#f**) does not make sense.

autoBeamCheck (procedure)

A procedure taking three arguments, *context*, *dir* [start/stop (-1 or 1)], and *test* [shortest note in the beam]. A non-**#f** return value starts or stops the auto beam.

autoBeaming (boolean)

If set to true then beams are generated automatically.

autoCautionaries (list)

List similar to **autoAccidentals**, but it controls cautionary accidentals rather than normal ones. Both lists are tried, and the one giving the most accidentals wins. In case of draw, a normal accidental is typeset.

automaticBars (boolean)

If set to false then bar lines will not be printed automatically; they must be explicitly created with a **\bar** command. Unlike the **\cadenzaOn** keyword, measures are still counted. Bar line generation will resume according to that count if this property is unset.

barAlways (boolean)

If set to true a bar line is drawn after each note.

barCheckSynchronize (boolean)

If true then reset **measurePosition** when finding a bar check.

barNumberFormatter (procedure)

A procedure that takes a bar number, measure position, and alternative number and returns a markup of the bar number to print.

barNumberVisibility (procedure)

A procedure that takes a bar number and a measure position and returns whether the corresponding bar number should be printed.

baseMoment (moment)

Smallest unit of time that will stand on its own as a subdivided section.

bassFigureFormatFunction (procedure)

A procedure that is called to produce the formatting for a **BassFigure** grob. It takes a list of **BassFigureEvents**, a context, and the grob to format.

bassStaffProperties (list)

An alist of property settings to apply for the down staff of **PianoStaff**. Used by `\autochange`.

beamExceptions (list)

An alist of exceptions to autobeam rules that normally end on beats.

beamHalfMeasure (boolean)

Whether to allow a beam to begin halfway through the measure in triple time, which could look like 6/8.

beatStructure (list)

List of **baseMoments** that are combined to make beats.

chordChanges (boolean)

Only show changes in chords scheme?

chordNameExceptions (list)

An alist of chord exceptions. Contains (*chord . markup*) entries.

chordNameExceptionsFull (list)

An alist of full chord exceptions. Contains (*chord . markup*) entries.

chordNameExceptionsPartial (list)

An alist of partial chord exceptions. Contains (*chord . (prefix-markup suffix-markup)*) entries.

chordNameFunction (procedure)

The function that converts lists of pitches to chord names.

chordNameLowercaseMinor (boolean)

Downcase roots of minor chords?

chordNameSeparator (markup)

The markup object used to separate parts of a chord name.

chordNoteNamer (procedure)

A function that converts from a pitch object to a text markup. Used for single pitches.

chordPrefixSpacer (number)

The space added between the root symbol and the prefix of a chord name.

chordRootNamer (procedure)

A function that converts from a pitch object to a text markup. Used for chords.

clefGlyph (string)

Name of the symbol within the music font.

clefOctavation (integer)

Add this much extra octavation. Values of 7 and -7 are common.

clefPosition (number)

Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

completionBusy (boolean)

Whether a completion-note head is playing.

completionUnit (moment)

Sub-bar unit of completion.

connectArpeggios (boolean)

If set, connect arpeggios across piano staff.

countPercentRepeats (boolean)

If set, produce counters for percent repeats.

createKeyOnClefChange (boolean)

Print a key signature whenever the clef is changed.

createSpacing (boolean)

Create **StaffSpacing** objects? Should be set for staves.

crescendoSpanner (symbol)

The type of spanner to be used for crescendi. Available values are ‘**hairpin**’ and ‘**text**’. If unset, a hairpin crescendo is used.

crescendoText (markup)

The text to print at start of non-hairpin crescendo, i.e., ‘**cresc.**’.

cueClefGlyph (string)

Name of the symbol within the music font.

cueClefOctavation (integer)

Add this much extra octavation. Values of 7 and -7 are common.

cueClefPosition (number)

Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

currentBarNumber (integer)

Contains the current barnumber. This property is incremented at every bar line.

decrescendoSpanner (symbol)

The type of spanner to be used for decrescendi. Available values are ‘**hairpin**’ and ‘**text**’. If unset, a hairpin decrescendo is used.

decrescendoText (markup)

The text to print at start of non-hairpin decrescendo, i.e., ‘**dim.**’.

defaultBarType (string)

Set the default type of bar line. See **whichBar** for information on available bar types. This variable is read by *Section “Timing-translator” dans Référence des propriétés internes* at *Section “Score” dans Référence des propriétés internes* level.

defaultStrings (list)

A list of strings to use in calculating frets for tablatures and fretboards if no strings are provided in the notes for the current moment.

doubleRepeatSegnoType (string)

Set the default bar line for the combinations double repeat with segno. Default is ‘**:|S|:**’.

doubleRepeatType (string)

Set the default bar line for double repeats.

doubleSlurs (boolean)

If set, two slurs are created for every slurred note, one above and one below the chord.

drumPitchTable (hash table)

A table mapping percussion instruments (symbols) to pitches.

drumStyleTable (hash table)

A hash table which maps drums to layout settings. Predefined values: ‘**drums-style**’, ‘**timbales-style**’, ‘**congas-style**’, ‘**bongos-style**’, and ‘**percussion-style**’.

The layout style is a hash table, containing the drum-pitches (e.g., the symbol ‘**hihat**’) as keys, and a list (***notehead-style script vertical-position***) as values.

endRepeatSegnoType (string)

Set the default bar line for the combinations ending of repeat with segno. Default is ‘:|S’.

endRepeatType (string)

Set the default bar line for the ending of repeats.

explicitClefVisibility (vector)

‘**break-visibility**’ function for clef changes.

explicitCueClefVisibility (vector)

‘**break-visibility**’ function for cue clef changes.

explicitKeySignatureVisibility (vector)

‘**break-visibility**’ function for explicit key changes. ‘**\override**’ of the **break-visibility** property will set the visibility for normal (i.e., at the start of the line) key signatures.

extendersOverRests (boolean)

Whether to continue extenders as they cross a rest.

extraNatural (boolean)

Whether to typeset an extra natural sign before accidentals that reduce the effect of a previous alteration.

figuredBassAlterationDirection (direction)

Where to put alterations relative to the main figure.

figuredBassCenterContinuations (boolean)

Whether to vertically center pairs of extender lines. This does not work with three or more lines.

figuredBassFormatter (procedure)

A routine generating a markup for a bass figure.

figuredBassPlusDirection (direction)

Where to put plus signs relative to the main figure.

fingeringOrientations (list)

A list of symbols, containing ‘**left**’, ‘**right**’, ‘**up**’ and/or ‘**down**’. This list determines where fingerings are put relative to the chord being fingered.

firstClef (boolean)

If true, create a new clef when starting a staff.

followVoice (boolean)

If set, note heads are tracked across staff switches by a thin line.

fontSize (number)

The relative size of all grobs in a context.

forbidBreak (boolean)

If set to **#t**, prevent a line break at this point.

forceClef (boolean)

Show clef symbol, even if it has not changed. Only active for the first clef after the property is set, not for the full staff.

fretLabels (list)

A list of strings or Scheme-formatted markups containing, in the correct order, the labels to be used for lettered frets in tablature.

glissandoMap (list)

A map in the form of '((source1 . target1) (source2 . target2) (sourcen . targetn)) showing the glissandi to be drawn for note columns. The value '() will default to '((0 . 0) (1 . 1) (n . n)), where n is the minimal number of note-heads in the two note columns between which the glissandi occur.

gridInterval (moment)

Interval for which to generate **GridPoints**.

handleNegativeFrets (symbol)

How the automatic fret calculator should handle calculated negative frets. Values include **'ignore**, to leave them out of the diagram completely, **'include**, to include them as calculated, and **'recalculate**, to ignore the specified string and find a string where they will fit with a positive fret number.

harmonicAccidentals (boolean)

If set, harmonic notes in chords get accidentals.

harmonicDots (boolean)

If set, harmonic notes in dotted chords get dots.

highStringOne (boolean)

Whether the first string is the string with highest pitch on the instrument. This used by the automatic string selector for tablature notation.

ignoreBarChecks (boolean)

Ignore bar checks.

ignoreFiguredBassRest (boolean)

Don't swallow rest events.

ignoreMelismata (boolean)

Ignore melismata for this **Section “Lyrics”** dans *Référence des propriétés internes* line.

implicitBassFigures (list)

A list of bass figures that are not printed as numbers, but only as extender lines.

implicitTimeSignatureVisibility (vector)

break visibility for the default time signature.

includeGraceNotes (boolean)

Do not ignore grace notes for **Section “Lyrics”** dans *Référence des propriétés internes*.

instrumentCueName (markup)

The name to print if another instrument is to be taken.

instrumentEqualizer (procedure)

A function taking a string (instrument name), and returning a (*min* . *max*) pair of numbers for the loudness range of the instrument.

instrumentName (markup)

The name to print left of a staff. The **instrumentName** property labels the staff in the first system, and the **shortInstrumentName** property labels following lines.

instrumentTransposition (pitch)

Define the transposition of the instrument. Its value is the pitch that sounds like middle C. This is used to transpose the MIDI output, and \quotes.

internalBarNumber (integer)

Contains the current barnumber. This property is used for internal timekeeping, among others by the **Accidental_engraver**.

keepAliveInterfaces (list)

A list of symbols, signifying grob interfaces that are worth keeping a staff with **remove-empty** set around for.

keyAlterationOrder (list)

An alist that defines in what order alterations should be printed. The format is (*step* . *alter*), where *step* is a number from 0 to 6 and *alter* from -2 (sharp) to 2 (flat).

keySignature (list)

The current key signature. This is an alist containing (*step* . *alter*) or ((*octave* . *step*) . *alter*), where *step* is a number in the range 0 to 6 and *alter* a fraction, denoting alteration. For alterations, use symbols, e.g. **keySignature** = #`((6 . ,FLAT)).

lyricMelismaAlignment (number)

Alignment to use for a melisma syllable.

majorSevenSymbol (markup)

How should the major 7th be formatted in a chord name?

markFormatter (procedure)

A procedure taking as arguments the context and the rehearsal mark. It should return the formatted mark as a markup object.

maximumFretStretch (number)

Don't allocate frets further than this from specified frets.

measureLength (moment)

Length of one measure in the current time signature.

measurePosition (moment)

How much of the current measure have we had. This can be set manually to create incomplete measures.

melismaBusyProperties (list)

A list of properties (symbols) to determine whether a melisma is playing. Setting this property will influence how lyrics are aligned to notes. For example, if set to '(**melismaBusy** **beamMelismaBusy**)', only manual melismata and manual beams are considered. Possible values include **melismaBusy**, **slurMelismaBusy**, **tieMelismaBusy**, and **beamMelismaBusy**.

metronomeMarkFormatter (procedure)

How to produce a metronome markup. Called with two arguments: a **TempoChangeEvent** and context.

middleCClefPosition (number)

The position of the middle C, as determined only by the clef. This can be calculated by looking at **clefPosition** and **clefGlyph**.

middleCCuePosition (number)

The position of the middle C, as determined only by the clef of the cue notes. This can be calculated by looking at **cueClefPosition** and **cueClefGlyph**.

middleCOffset (number)

The offset of middle C from the position given by **middleCClefPosition**. This is used for ottava brackets.

middleCPosition (number)

The place of the middle C, measured in half staff-spaces. Usually determined by looking at **middleCClefPosition** and **middleCOffset**.

midiChannelMapping (symbol)

How to map MIDI channels: per **instrument** (default), **staff** or **voice**.

midiInstrument (string)

Name of the MIDI instrument to use.

midiMaximumVolume (number)

Analogous to **midiMinimumVolume**.

midiMergeUnisons (boolean)

If true, output only one MIDI note-on event when notes with the same pitch, in the same MIDI-file track, overlap.

midiMinimumVolume (number)

Set the minimum loudness for MIDI. Ranges from 0 to 1.

minimumFret (number)

The tablature auto string-selecting mechanism selects the highest string with a fret at least **minimumFret**.

minimumPageTurnLength (moment)

Minimum length of a rest for a page turn to be allowed.

minimumRepeatLengthForPageTurn (moment)

Minimum length of a repeated section for a page turn to be allowed within that section.

minorChordModifier (markup)

Markup displayed following the root for a minor chord

noChordSymbol (markup)

Markup to be displayed for rests in a **ChordNames** context.

noteToFretFunction (procedure)

Convert list of notes and list of defined strings to full list of strings and fret numbers. Parameters: The context, a list of note events, a list of tabstring events, and the fretboard grob if a fretboard is desired.

ottavation (markup)

If set, the text for an ottava spanner. Changing this creates a new text spanner.

`output` (music output)

The output produced by a score-level translator during music interpretation.

`partCombineTextsOnNote` (boolean)

Print part-combine texts only on the next note rather than immediately on rests or skips.

`pedalSostenutoStrings` (list)

See `pedalSustainStrings`.

`pedalSostenutoStyle` (symbol)

See `pedalSustainStyle`.

`pedalSustainStrings` (list)

A list of strings to print for sustain-pedal. Format is (*up updown down*), where each of the three is the string to print when this is done with the pedal.

`pedalSustainStyle` (symbol)

A symbol that indicates how to print sustain pedals: `text`, `bracket` or `mixed` (both).

`pedalUnaCordaStrings` (list)

See `pedalSustainStrings`.

`pedalUnaCordaStyle` (symbol)

See `pedalSustainStyle`.

`predefinedDiagramTable` (hash table)

The hash table of predefined fret diagrams to use in `FretBoards`.

`printKeyCancellation` (boolean)

Print restoration alterations before a key signature change.

`printOctaveNames` (boolean)

Print octave marks for the `NoteNames` context.

`printPartCombineTexts` (boolean)

Set 'Solo' and 'A due' texts in the part combiner?

`proportionalNotationDuration` (moment)

Global override for shortest-playing duration. This is used for switching on proportional notation.

`rehearsalMark` (integer)

The last rehearsal mark printed.

`repeatCommands` (list)

This property is a list of commands of the form (`list 'volta x`), where `x` is a string or `#f`. 'end-repeat' is also accepted as a command.

`repeatCountVisibility` (procedure)

A procedure taking as arguments an integer and context, returning whether the corresponding percent repeat number should be printed when `countPercentRepeats` is set.

`restCompletionBusy` (boolean)

Signal whether a completion-rest is active.

`restNumberThreshold` (number)

If a multimeasure rest has more measures than this, a number is printed.

`restrainOpenStrings` (boolean)

Exclude open strings from the automatic fret calculator.

searchForVoice (boolean)

Signal whether a search should be made of all contexts in the context hierarchy for a voice to provide rhythms for the lyrics.

segnoType (string)

Set the default bar line for a requested segno. Default is ‘S’.

shapeNoteStyles (vector)

Vector of symbols, listing style for each note head relative to the tonic (qv.) of the scale.

shortInstrumentName (markup)

See **instrumentName**.

shortVocalName (markup)

Name of a vocal line, short version.

skipBars (boolean)

If set to true, then skip the empty bars that are produced by multimeasure notes and rests. These bars will not appear on the printed output. If not set (the default), multimeasure notes and rests expand into their full length, printing the appropriate number of empty bars so that synchronization with other voices is preserved.

```
{
  r1 r1*3 R1*3
  \set Score.skipBars= ##t
  r1*3 R1*3
}
```

skipTypesetting (boolean)

If true, no typesetting is done, speeding up the interpretation phase. Useful for debugging large scores.

slashChordSeparator (markup)

The markup object used to separate a chord name from its root note in case of inversions or slash chords.

soloIIText (markup)

The text for the start of a solo for voice ‘two’ when part-combining.

soloText (markup)

The text for the start of a solo when part-combining.

squashedPosition (integer)

Vertical position of squashing for **Section “Pitch_squash_engraver”** dans *Référence des propriétés internes*.

staffLineLayoutFunction (procedure)

Layout of staff lines, **traditional**, or **semitone**.

stanza (markup)

Stanza ‘number’ to print before the start of a verse. Use in **Lyrics** context.

startRepeatSegnoType (string)

Set the default bar line for the combinations beginning of repeat with segno. Default is ‘.S|:’.

startRepeatType (string)

Set the default bar line for the beginning of repeats.

- stemLeftBeamCount** (integer)
Specify the number of beams to draw on the left side of the next note. Overrides automatic beaming. The value is only used once, and then it is erased.
- stemRightBeamCount** (integer)
See **stemLeftBeamCount**.
- strictBeatBeaming** (boolean)
Should partial beams reflect the beat structure even if it causes flags to hang out?
- stringNumberOrientations** (list)
See **fingeringOrientations**.
- stringOneTopmost** (boolean)
Whether the first string is printed on the top line of the tablature.
- stringTunings** (list)
The tablature strings tuning. It is a list of the pitches of each string (starting with the lowest numbered one).
- strokeFingerOrientations** (list)
See **fingeringOrientations**.
- subdivideBeams** (boolean)
If set, multiple beams will be subdivided at **baseMoment** positions by only drawing one beam over the beat.
- suggestAccidentals** (boolean)
If set, accidentals are typeset as cautionary suggestions over the note.
- systemStartDelimiter** (symbol)
Which grob to make for the start of the system/staff? Set to **SystemStartBrace**, **SystemStartBracket** or **SystemStartBar**.
- systemStartDelimiterHierarchy** (pair)
A nested list, indicating the nesting of a start delimiters.
- tablatureFormat** (procedure)
A function formatting a tablature note head. Called with three arguments: context, string number and, fret number. It returns the text as a markup.
- tabStaffLineLayoutFunction** (procedure)
A function determining the staff position of a tablature note head. Called with two arguments: the context and the string.
- tempoHideNote** (boolean)
Hide the note = count in tempo marks.
- tempoWholesPerMinute** (moment)
The tempo in whole notes per minute.
- tieWaitForNote** (boolean)
If true, tied notes do not have to follow each other directly. This can be used for writing out arpeggios.
- timeSignatureFraction** (fraction, as pair)
A pair of numbers, signifying the time signature. For example, '(4 . 4)' is a 4/4 time signature.
- timeSignatureSettings** (list)
A nested alist of settings for time signatures. Contains elements for various time signatures. The element for each time signature contains entries for **baseMoment**, **beatStructure**, and **beamExceptions**.

timing (boolean)

Keep administration of measure length, position, bar number, etc.? Switch off for cadenzas.

tonic (pitch)

The tonic of the current scale.

topLevelAlignment (boolean)

If true, the *Vertical-align-engraver* will create a *VerticalAlignment*; otherwise, it will create a *StaffGrouper*

trebleStaffProperties (list)

An alist of property settings to apply for the up staff of *PianoStaff*. Used by `\autochange`.

tremoloFlags (integer)

The number of tremolo flags to add if no number is specified.

tupletFullLength (boolean)

If set, the tuplet is printed up to the start of the next note.

tupletFullLengthNote (boolean)

If set, end at the next note, otherwise end on the matter (time signatures, etc.) before the note.

tupletSpannerDuration (moment)

Normally, a tuplet bracket is as wide as the `\times` expression that gave rise to it. By setting this property, you can make brackets last shorter.

```
{
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```

useBassFigureExtenders (boolean)

Whether to use extender lines for repeated bass figures.

vocalName (markup)

Name of a vocal line.

voltaSpannerDuration (moment)

This specifies the maximum duration to use for the brackets printed for `\alternative`. This can be used to shrink the length of brackets in the situation where one alternative is very large.

whichBar (string)

This property is read to determine what type of bar line to create.

Example:

```
\set Staff.whichBar = "|: "
```

This will create a start-repeat bar in this staff only. Valid values are described in [Section “bar-line-interface”](#) dans *Référence des propriétés internes*.

A.17 Propriétés de mise en forme

add-stem-support (boolean)

If set, the *Stem* object is included in this script's support.

after-line-breaking (boolean)

Dummy property, used to trigger callback for `after-line-breaking`.

- align-dir** (direction)
Which side to align? -1: left side, 0: around center of width, 1: right side.
- allow-loose-spacing** (boolean)
If set, column can be detached from main spacing.
- allow-span-bar** (boolean)
If false, no inter-staff bar line will be created below this bar line.
- alteration** (number)
Alteration numbers for accidental.
- alteration-alist** (list)
List of (*pitch* . *accidental*) pairs for key signature.
- annotation** (string)
Annotate a grob for debug purposes.
- annotation-balloon** (boolean)
Print the balloon around an annotation.
- annotation-line** (boolean)
Print the line from an annotation to the grob that it annotates.
- arpeggio-direction** (direction)
If set, put an arrow on the arpeggio squiggly line.
- arrow-length** (number)
Arrow length.
- arrow-width** (number)
Arrow width.
- auto-knee-gap** (dimension, in staff space)
If a gap is found between note heads where a horizontal beam fits that is larger than this number, make a kneed beam.
- automatically-numbered** (boolean)
Should a footnote be automatically numbered?
- average-spacing-wishes** (boolean)
If set, the spacing wishes are averaged over staves.
- avoid-note-head** (boolean)
If set, the stem of a chord does not pass through all note heads, but starts at the last note head.
- avoid-scripts** (boolean)
If set, a tuplet bracket avoids the scripts associated with the note heads it encompasses.
- avoid-slur** (symbol)
Method of handling slur collisions. Choices are **inside**, **outside**, **around**, and **ignore**. **inside** adjusts the slur if needed to keep the grob inside the slur. **outside** moves the grob vertically to the outside of the slur. **around** moves the grob vertically to the outside of the slur only if there is a collision. **ignore** does not move either. In grobs whose notational significance depends on vertical position (such as accidentals, clefs, etc.), **outside** and **around** behave like **ignore**.
- axes** (list) List of axis numbers. In the case of alignment grobs, this should contain only one number.

bar-extent (pair of numbers)

The Y-extent of the actual bar line. This may differ from **Y-extent** because it does not include the dots in a repeat bar line.

base-shortest-duration (moment)

Spacing is based on the shortest notes in a piece. Normally, pieces are spaced as if notes at least as short as this are present.

baseline-skip (dimension, in staff space)

Distance between base lines of multiple lines of text.

beam-thickness (dimension, in staff space)

Beam thickness, measured in **staff-space** units.

beam-width (dimension, in staff space)

Width of the tremolo sign.

beamed-stem-shorten (list)

How much to shorten beamed stems, when their direction is forced. It is a list, since the value is different depending on the number of flags and beams.

beaming (pair)

Pair of number lists. Each number list specifies which beams to make. 0 is the central beam, 1 is the next beam toward the note, etc. This information is used to determine how to connect the beaming patterns from stem to stem inside a beam.

beamlet-default-length (pair)

A pair of numbers. The first number specifies the default length of a beamlet that sticks out of the left hand side of this stem; the second number specifies the default length of the beamlet to the right. The actual length of a beamlet is determined by taking either the default length or the length specified by **beamlet-max-length-proportion**, whichever is smaller.

beamlet-max-length-proportion (pair)

The maximum length of a beamlet, as a proportion of the distance between two adjacent stems.

before-line-breaking (boolean)

Dummy property, used to trigger a callback function.

between-cols (pair)

Where to attach a loose column to.

bound-details (list)

An alist of properties for determining attachments of spanners to edges.

bound-padding (number)

The amount of padding to insert around spanner bounds.

bracket-flare (pair of numbers)

A pair of numbers specifying how much edges of brackets should slant outward. Value 0.0 means straight edges.

bracket-visibility (boolean or symbol)

This controls the visibility of the tuplet bracket. Setting it to false prevents printing of the bracket. Setting the property to **if-no-beam** makes it print only if there is no beam associated with this tuplet bracket.

break-align-anchor (number)

Grobs aligned to this break-align grob will have their X-offsets shifted by this number. In bar lines, for example, this is used to position grobs relative to the (visual) center of the bar line.

break-align-anchor-alignment (number)

Read by `ly:break-aligned-interface::calc-extent-aligned-anchor` for aligning an anchor to a grob's extent.

break-align-orders (vector)

Defines the order in which prefatory matter (clefs, key signatures) appears. The format is a vector of length 3, where each element is one order for end-of-line, middle of line, and start-of-line, respectively. An order is a list of symbols.

For example, clefs are put after key signatures by setting

```
\override Score.BreakAlignment #'break-align-orders =
  #(make-vector 3 '(span-bar
                    breathing-sign
                    staff-bar
                    key
                    clef
                    time-signature))
```

break-align-symbol (symbol)

This key is used for aligning and spacing breakable items.

break-align-symbols (list)

A list of symbols that determine which break-aligned grobs to align this to. If the grob selected by the first symbol in the list is invisible due to break-visibility, we will align to the next grob (and so on). Choices are `left-edge`, `ambitus`, `breathing-sign`, `clef`, `staff-bar`, `key-cancellation`, `key-signature`, `time-signature`, and `custos`.

break-overshoot (pair of numbers)

How much does a broken spanner stick out of its bounds?

break-visibility (vector)

A vector of 3 booleans, `#(end-of-line unbroken begin-of-line)`. `#t` means visible, `#f` means killed.

breakable (boolean)

Allow breaks here.

broken-bound-padding (number)

The amount of padding to insert when a spanner is broken at a line break.

c0-position (integer)

An integer indicating the position of middle C.

circled-tip (boolean)

Put a circle at start/end of hairpins (`al/del niente`).

clip-edges (boolean)

Allow outward pointing beamlets at the edges of beams?

collapse-height (dimension, in staff space)

Minimum height of system start delimiter. If equal or smaller, the bracket/brace/line is removed.

collision-bias (number)

Number determining how much to favor the left (negative) or right (positive). Larger absolute values in either direction will push a collision in this direction.

collision-interfaces (list)

A list of interfaces for which automatic beam-collision resolution is run.

collision-padding (number)

Amount of padding to apply after a collision is detected via the self-alignment-interface.

collision-voice-only (boolean)

Does automatic beam collision apply only to the voice in which the beam was created?

color (color)

The color of this grob.

common-shortest-duration (moment)

The most common shortest note length. This is used in spacing. Enlarging this sets the score tighter.

concaveness (number)

A beam is concave if its inner stems are closer to the beam than the two outside stems. This number is a measure of the closeness of the inner stems. It is used for damping the slope of the beam.

connect-to-neighbor (pair)

Pair of booleans, indicating whether this grob looks as a continued break.

control-points (list)

List of offsets (number pairs) that form control points for the tie, slur, or bracket shape. For Béziers, this should list the control points of a third-order Bézier curve.

damping (number)

Amount of beam slope damping.

dash-definition (pair)

List of **dash-elements** defining the dash structure. Each **dash-element** has a starting t value, an ending t-value, a **dash-fraction**, and a **dash-period**.

dash-fraction (number)

Size of the dashes, relative to **dash-period**. Should be between 0.0 (no line) and 1.0 (continuous line).

dash-period (number)

The length of one dash together with whitespace. If negative, no line is drawn at all.

default-direction (direction)

Direction determined by note head positions.

default-staff-staff-spacing (list)

The settings to use for **staff-staff-spacing** when it is unset, for ungrouped staves and for grouped staves that do not have the relevant **StaffGrouper** property set (**staff-staff-spacing** or **staffgroup-staff-spacing**).

details (list)

Alist of parameters for detailed grob behavior. More information on the allowed parameters for a grob can be found by looking at the top of the Internals Reference page for each interface having a **details** property.

digit-names (vector)

Names for string finger digits.

direction (direction)

If **side-axis** is 0 (or X), then this property determines whether the object is placed LEFT, CENTER or RIGHT with respect to the other object. Otherwise, it determines

whether the object is placed UP, CENTER or DOWN. Numerical values may also be used: UP=1, DOWN=-1, LEFT=-1, RIGHT=1, CENTER=0.

dot-count (integer)

The number of dots.

dot-negative-kern (number)

The space to remove between a dot and a slash in percent repeat glyphs. Larger values bring the two elements closer together.

dot-placement-list (list)

List consisting of (*description string-number fret-number finger-number*) entries used to define fret diagrams.

duration-log (integer)

The 2-log of the note head duration, i.e., 0 = whole note, 1 = half note, etc.

eccentricity (number)

How asymmetrical to make a slur. Positive means move the center to the right.

edge-height (pair)

A pair of numbers specifying the heights of the vertical edges: (*left-height . right-height*).

edge-text (pair)

A pair specifying the texts to be set at the edges: (*left-text . right-text*).

expand-limit (integer)

Maximum number of measures expanded in church rests.

extra-dy (number)

Slope glissandi this much extra.

extra-offset (pair of numbers)

A pair representing an offset. This offset is added just before outputting the symbol, so the typesetting engine is completely oblivious to it. The values are measured in **staff-space** units of the staff's **StaffSymbol**.

extra-spacing-height (pair of numbers)

In the horizontal spacing problem, we increase the height of each item by this amount (by adding the 'car' to the bottom of the item and adding the 'cdr' to the top of the item). In order to make a grob infinitely high (to prevent the horizontal spacing problem from placing any other grobs above or below this grob), set this to (**-inf.0 . +inf.0**).

extra-spacing-width (pair of numbers)

In the horizontal spacing problem, we pad each item by this amount (by adding the 'car' on the left side of the item and adding the 'cdr' on the right side of the item). In order to make a grob take up no horizontal space at all, set this to (**+inf.0 . -inf.0**).

extra-X-extent (pair of numbers)

A grob is enlarged in X dimension by this much.

extra-Y-extent (pair of numbers)

A grob is enlarged in Y dimension by this much.

flag-count (number)

The number of tremolo beams.

font-encoding (symbol)

The font encoding is the broadest category for selecting a font. Currently, only lilypond's system fonts (Emmentaler) are using this property. Available values are **fetaMusic** (Emmentaler), **fetaBraces**, **fetaText** (Emmentaler).

font-family (symbol)

The font family is the broadest category for selecting text fonts. Options include: **sans**, **roman**.

font-name (string)

Specifies a file name (without extension) of the font to load. This setting overrides selection using **font-family**, **font-series** and **font-shape**.

font-series (symbol)

Select the series of a font. Choices include **medium**, **bold**, **bold-narrow**, etc.

font-shape (symbol)

Select the shape of a font. Choices include **upright**, **italic**, **caps**.

font-size (number)

The font size, compared to the 'normal' size. 0 is style-sheet's normal size, -1 is smaller, +1 is bigger. Each step of 1 is approximately 12% larger; 6 steps are exactly a factor 2 larger. Fractional values are allowed.

footnote (boolean)

Should this be a footnote or in-note?

footnote-music (music)

Music creating a footnote.

footnote-text (markup)

A footnote for the grob.

force-hshift (number)

This specifies a manual shift for notes in collisions. The unit is the note head width of the first voice note. This is used by [Section “note-collision-interface”](#) dans *Référence des propriétés internes*.

forced-spacing (number)

Spacing forced between grobs, used in various ligature engravers.

fraction (fraction, as pair)

Numerator and denominator of a time signature object.

french-beaming (boolean)

Use French beaming style for this stem. The stem stops at the innermost beams.

fret-diagram-details (list)

An alist of detailed grob properties for fret diagrams. Each alist entry consists of a (*property* . *value*) pair. The properties which can be included in **fret-diagram-details** include the following:

- **barre-type** – Type of barre indication used. Choices include **curved**, **straight**, and **none**. Default **curved**.
- **capo-thickness** – Thickness of capo indicator, in multiples of fret-space. Default value 0.5.
- **dot-color** – Color of dots. Options include **black** and **white**. Default **black**.
- **dot-label-font-mag** – Magnification for font used to label fret dots. Default value 1.

- **dot-position** – Location of dot in fret space. Default 0.6 for dots without labels, 0.95-dot-radius for dots with labels.
- **dot-radius** – Radius of dots, in terms of fret spaces. Default value 0.425 for labeled dots, 0.25 for unlabeled dots.
- **finger-code** – Code for the type of fingering indication used. Options include **none**, **in-dot**, and **below-string**. Default **none** for markup fret diagrams, **below-string** for FretBoards fret diagrams.
- **fret-count** – The number of frets. Default 4.
- **fret-label-custom-format** – The format string to be used label the lowest fret number, when **number-type** equals to **custom**. Default "~a".
- **fret-label-font-mag** – The magnification of the font used to label the lowest fret number. Default 0.5.
- **fret-label-vertical-offset** – The offset of the fret label from the center of the fret in direction parallel to strings. Default 0.
- **label-dir** – Side to which the fret label is attached. -1, **LEFT**, or **DOWN** for left or down; 1, **RIGHT**, or **UP** for right or up. Default **RIGHT**.
- **mute-string** – Character string to be used to indicate muted string. Default "x".
- **number-type** – Type of numbers to use in fret label. Choices include **roman-lower**, **roman-upper**, **arabic** and **custom**. In the later case, the format string is supplied by the **fret-label-custom-format** property. Default **roman-lower**.
- **open-string** – Character string to be used to indicate open string. Default "o".
- **orientation** – Orientation of fret-diagram. Options include **normal**, **landscape**, and **opposing-landscape**. Default **normal**.
- **string-count** – The number of strings. Default 6.
- **string-label-font-mag** – The magnification of the font used to label fingerings at the string, rather than in the dot. Default value 0.6 for **normal** orientation, 0.5 for **landscape** and **opposing-landscape**.
- **string-thickness-factor** – Factor for changing thickness of each string in the fret diagram. Thickness of string k is given by $\text{thickness} * (1 + \text{string-thickness-factor})^{(k-1)}$. Default 0.
- **top-fret-thickness** – The thickness of the top fret line, as a multiple of the standard thickness. Default value 3.
- **xo-font-magnification** – Magnification used for mute and open string indicators. Default value 0.5.
- **xo-padding** – Padding for open and mute indicators from top fret. Default value 0.25.

full-length-padding (number)

How much padding to use at the right side of a full-length tuplet bracket.

full-length-to-extent (boolean)

Run to the extent of the column for a full-length tuplet bracket.

full-measure-extra-space (number)

Extra space that is allocated at the beginning of a measure with only one note. This property is read from the `NonMusicalPaperColumn` that begins the measure.

full-size-change (boolean)

Don't make a change clef smaller.

- gap** (dimension, in staff space)
Size of a gap in a variable symbol.
- gap-count** (integer)
Number of gapped beams for tremolo.
- glissando-skip** (boolean)
Should this **NoteHead** be skipped by glissandi?
- glyph** (string)
A string determining what ‘style’ of glyph is typeset. Valid choices depend on the function that is reading this property.
- glyph-name** (string)
The glyph name within the font.
- glyph-name-alist** (list)
An alist of key-string pairs.
- graphical** (boolean)
Display in graphical (vs. text) form.
- grow-direction** (direction)
Crescendo or decrescendo?
- hair-thickness** (number)
Thickness of the thin line in a bar line.
- harp-pedal-details** (list)
An alist of detailed grob properties for harp pedal diagrams. Each alist entry consists of a (*property* . *value*) pair. The properties which can be included in harp-pedal-details include the following:
- **box-offset** – Vertical shift of the center of flat/sharp pedal boxes above/below the horizontal line. Default value 0.8.
 - **box-width** – Width of each pedal box. Default value 0.4.
 - **box-height** – Height of each pedal box. Default value 1.0.
 - **space-before-divider** – Space between boxes before the first divider (so that the diagram can be made symmetric). Default value 0.8.
 - **space-after-divider** – Space between boxes after the first divider. Default value 0.8.
 - **circle-thickness** – Thickness (in unit of the line-thickness) of the ellipse around circled pedals. Default value 0.5.
 - **circle-x-padding** – Padding in X direction of the ellipse around circled pedals. Default value 0.15.
 - **circle-y-padding** – Padding in Y direction of the ellipse around circled pedals. Default value 0.2.
- head-direction** (direction)
Are the note heads left or right in a semitie?
- height** (dimension, in staff space)
Height of an object in **staff-space** units.
- height-limit** (dimension, in staff space)
Maximum slur height: The longer the slur, the closer it is to this height.
- hide-tied-accidental-after-break** (boolean)
If set, an accidental that appears on a tied note after a line break will not be displayed.

horizontal-shift (integer)

An integer that identifies ranking of `NoteColumns` for horizontal shifting. This is used by [Section “note-collision-interface”](#) dans *Référence des propriétés internes*.

horizontal-skylines (pair of skylines)

Two skylines, one to the left and one to the right of this grob.

id (string)

An id string for the grob. Depending on the typesetting backend being used, this id will be assigned to a group containing all of the stencils that comprise a given grob. For example, in the svg backend, the string will be assigned to the `id` attribute of a group (`<g>`) that encloses the stencils that comprise the grob. In the Postscript backend, as there is no way to group items, the setting of the `id` property will have no effect.

ignore-collision (boolean)

If set, don't do note collision resolution on this `NoteColumn`.

implicit (boolean)

Is this an implicit bass figure?

inspect-index (integer)

If debugging is set, set beam and slur configuration to this index, and print the respective scores.

inspect-quants (pair of numbers)

If debugging is set, set beam and slur quants to this position, and print the respective scores.

keep-inside-line (boolean)

If set, this column cannot have objects sticking into the margin.

kern (dimension, in staff space)

Amount of extra white space to add. For bar lines, this is the amount of space after a thick line.

knee (boolean)

Is this beam kneed?

knee-spacing-correction (number)

Factor for the optical correction amount for kneed beams. Set between 0 for no correction and 1 for full correction.

labels (list)

List of labels (symbols) placed on a column.

layer (integer)

An integer which determines the order of printing objects. Objects with the lowest value of layer are drawn first, then objects with progressively higher values are drawn, so objects with higher values overwrite objects with lower values. By default most objects are assigned a layer value of 1.

ledger-extra (dimension, in staff space)

Extra distance from staff line to draw ledger lines for.

ledger-line-thickness (pair of numbers)

The thickness of ledger lines. It is the sum of 2 numbers: The first is the factor for line thickness, and the second for staff space. Both contributions are added.

ledger-positions (list)

Repeating pattern for the vertical positions of ledger lines. Bracketed groups are always shown together.

left-bound-info (list)

An alist of properties for determining attachments of spanners to edges.

left-padding (dimension, in staff space)

The amount of space that is put left to an object (e.g., a lyric extender).

length (dimension, in staff space)

User override for the stem length of unbeamed stems.

length-fraction (number)

Multiplier for lengths. Used for determining ledger lines and stem lengths.

line-break-penalty (number)

Penalty for a line break at this column. This affects the choices of the line breaker; it avoids a line break at a column with a positive penalty and prefers a line break at a column with a negative penalty.

line-break-permission (symbol)

Instructs the line breaker on whether to put a line break at this column. Can be **force** or **allow**.

line-break-system-details (list)

An alist of properties to use if this column is the start of a system.

line-count (integer)

The number of staff lines.

line-positions (list)

Vertical positions of staff lines.

line-thickness (number)

The thickness of the tie or slur contour.

long-text (markup)

Text markup. See [Section “Formatting text” dans *Manuel de notation*](#).

max-beam-connect (integer)

Maximum number of beams to connect to beams from this stem. Further beams are typeset as beamlets.

max-stretch (number)

The maximum amount that this `VerticalAxisGroup` can be vertically stretched (for example, in order to better fill a page).

measure-count (integer)

The number of measures for a multi-measure rest.

measure-length (moment)

Length of a measure. Used in some spacing situations.

merge-differently-dotted (boolean)

Merge note heads in collisions, even if they have a different number of dots. This is normal notation for some types of polyphonic music.

merge-differently-dotted only applies to opposing stem directions (i.e., voice 1 & 2).

merge-differently-headed (boolean)

Merge note heads in collisions, even if they have different note heads. The smaller of the two heads is rendered invisible. This is used in polyphonic guitar notation. The value of this setting is used by Section “note-collision-interface” dans *Référence des propriétés internes*.

merge-differently-headed only applies to opposing stem directions (i.e., voice 1 & 2).

minimum-distance (dimension, in staff space)

Minimum distance between rest and notes or beam.

minimum-length (dimension, in staff space)

Try to make a spanner at least this long, normally in the horizontal direction. This requires an appropriate callback for the **springs-and-rods** property. If added to a **Tie**, this sets the minimum distance between noteheads.

minimum-length-fraction (number)

Minimum length of ledger line as fraction of note head size.

minimum-space (dimension, in staff space)

Minimum distance that the victim should move (after padding).

minimum-X-extent (pair of numbers)

Minimum size of an object in X dimension, measured in **staff-space** units.

minimum-Y-extent (pair of numbers)

Minimum size of an object in Y dimension, measured in **staff-space** units.

neutral-direction (direction)

Which direction to take in the center of the staff.

neutral-position (number)

Position (in half staff spaces) where to flip the direction of custos stem.

next (graphical (layout) object)

Object that is next relation (e.g., the lyric syllable following an extender).

no-alignment (boolean)

If set, don't place this grob in a **VerticalAlignment**; rather, place it using its own **Y-offset** callback.

no-ledgers (boolean)

If set, don't draw ledger lines on this object.

no-stem-extend (boolean)

If set, notes with ledger lines do not get stems extending to the middle staff line.

non-break-align-symbols (list)

A list of symbols that determine which NON-break-aligned interfaces to align this to.

non-default (boolean)

Set for manually specified clefs.

non-musical (boolean)

True if the grob belongs to a **NonMusicalPaperColumn**.

nonstaff-nonstaff-spacing (list)

The spacing alist controlling the distance between the current non-staff line and the next non-staff line in the direction of **staff-affinity**, if both are on the same side of the related staff, and **staff-affinity** is either UP or DOWN. See **staff-staff-spacing** for a description of the alist structure.

nonstaff-relatedstaff-spacing (list)

The spacing alist controlling the distance between the current non-staff line and the nearest staff in the direction of **staff-affinity**, if there are no non-staff lines between the two, and **staff-affinity** is either UP or DOWN. If **staff-affinity** is CENTER, then **nonstaff-relatedstaff-spacing** is used for the nearest staves on *both* sides, even if other non-staff lines appear between the current one and either of the staves. See **staff-staff-spacing** for a description of the alist structure.

nonstaff-unrelatedstaff-spacing (list)

The spacing alist controlling the distance between the current non-staff line and the nearest staff in the opposite direction from **staff-affinity**, if there are no other non-staff lines between the two, and **staff-affinity** is either UP or DOWN. See **staff-staff-spacing** for a description of the alist structure.

normalized-endpoints (pair)

Represents left and right placement over the total spanner, where the width of the spanner is normalized between 0 and 1.

note-names (vector)

Vector of strings containing names for easy-notation note heads.

outside-staff-horizontal-padding (number)

By default, an outside-staff-object can be placed so that it is very close to another grob horizontally. If this property is set, the outside-staff-object is raised so that it is not so close to its neighbor.

outside-staff-padding (number)

The padding to place between this grob and the staff when spacing according to **outside-staff-priority**.

outside-staff-priority (number)

If set, the grob is positioned outside the staff in such a way as to avoid all collisions. In case of a potential collision, the grob with the smaller **outside-staff-priority** is closer to the staff.

packed-spacing (boolean)

If set, the notes are spaced as tightly as possible.

padding (dimension, in staff space)

Add this much extra space between objects that are next to each other.

padding-pairs (list)

An alist mapping (*name* . *name*) to distances.

page-break-penalty (number)

Penalty for page break at this column. This affects the choices of the page breaker; it avoids a page break at a column with a positive penalty and prefers a page break at a column with a negative penalty.

page-break-permission (symbol)

Instructs the page breaker on whether to put a page break at this column. Can be **force** or **allow**.

page-turn-penalty (number)

Penalty for a page turn at this column. This affects the choices of the page breaker; it avoids a page turn at a column with a positive penalty and prefers a page turn at a column with a negative penalty.

- page-turn-permission** (symbol)
Instructs the page breaker on whether to put a page turn at this column. Can be **force** or **allow**.
- parenthesized** (boolean)
Parenthesize this grob.
- positions** (pair of numbers)
Pair of staff coordinates (*left* . *right*), where both *left* and *right* are in **staff-space** units of the current staff. For slurs, this value selects which slur candidate to use; if extreme positions are requested, the closest one is taken.
- prefer-dotted-right** (boolean)
For note collisions, prefer to shift dotted up-note to the right, rather than shifting just the dot.
- ratio** (number)
Parameter for slur shape. The higher this number, the quicker the slur attains its **height-limit**.
- remove-empty** (boolean)
If set, remove group if it contains no interesting items.
- remove-first** (boolean)
Remove the first staff of an orchestral score?
- replacement-alist** (list)
Alist of strings. The key is a string of the pattern to be replaced. The value is a string of what should be displayed. Useful for ligatures.
- restore-first** (boolean)
Print a natural before the accidental.
- rhythmic-location** (rhythmic location)
Where (bar number, measure position) in the score.
- right-bound-info** (list)
An alist of properties for determining attachments of spanners to edges.
- right-padding** (dimension, in staff space)
Space to insert on the right side of an object (e.g., between note and its accidentals).
- rotation** (list)
Number of degrees to rotate this object, and what point to rotate around. For example, '(45 0 0) rotates by 45 degrees around the center of this object.
- round-up-exceptions** (list)
A list of pairs where car is the numerator and cdr the denominator of a moment. Each pair in this list means that the multi-measure rests of the corresponding length will be rounded up to the longer rest. See *round-up-to-longer-rest*.
- round-up-to-longer-rest** (boolean)
Displays the longer multi-measure rest when the length of a measure is between two values of **usable-duration-logs**. For example, displays a breve instead of a whole in a 3/2 measure.
- same-direction-correction** (number)
Optical correction amount for stems that are placed in tight configurations. This amount is used for stems with the same direction to compensate for note head to stem distance.

script-priority (number)

A sorting key that determines in what order a script is within a stack of scripts.

self-alignment-X (number)

Specify alignment of an object. The value -1 means left aligned, 0 centered, and 1 right-aligned in X direction. Other numerical values may also be specified.

self-alignment-Y (number)

Like **self-alignment-X** but for the Y axis.

shorten-pair (pair of numbers)

The lengths to shorten a text-spanner on both sides, for example a pedal bracket. Positive values shorten the text-spanner, while negative values lengthen it.

shortest-duration-space (dimension, in staff space)

Start with this much space for the shortest duration. This is expressed in **spacing-increment** as unit. See also [Section “spacing-spanner-interface”](#) dans *Référence des propriétés internes*.

shortest-playing-duration (moment)

The duration of the shortest note playing here.

shortest-starter-duration (moment)

The duration of the shortest note that starts here.

side-axis (number)

If the value is X (or equivalently 0), the object is placed horizontally next to the other object. If the value is Y or 1, it is placed vertically.

side-relative-direction (direction)

Multiply direction of **direction-source** with this to get the direction of this object.

simple-Y (boolean)

Should the Y placement of a spanner disregard changes in system heights?

size (number)

Size of object, relative to standard size.

skip-quanting (boolean)

Should beam quanting be skipped?

skyline-horizontal-padding (number)

For determining the vertical distance between two staves, it is possible to have a configuration which would result in a tight interleaving of grobs from the top staff and the bottom staff. The larger this parameter is, the farther apart the staves are placed in such a configuration.

skyline-vertical-padding (number)

The amount by which the left and right skylines of a column are padded vertically, beyond the **Y-extents** and **extra-spacing-heights** of the constituent grobs in the column. Increase this to prevent interleaving of grobs from adjacent columns.

slash-negative-kern (number)

The space to remove between slashes in percent repeat glyphs. Larger values bring the two elements closer together.

slope (number)

The slope of this object.

slur-padding (number)

Extra distance between slur and script.

space-alist (list)

A table that specifies distances between prefatory items, like clef and time-signature. The format is an alist of spacing tuples: (*break-align-symbol type . distance*), where *type* can be the symbols `minimum-space` or `extra-space`.

space-to-barline (boolean)

If set, the distance between a note and the following non-musical column will be measured to the bar line instead of to the beginning of the non-musical column. If there is a clef change followed by a bar line, for example, this means that we will try to space the non-musical column as though the clef is not there.

spacing-increment (number)

Add this much space for a doubled duration. Typically, the width of a note head. See also [Section “spacing-spanner-interface”](#) dans *Référence des propriétés internes*.

spacing-pair (pair)

A pair of alignment symbols which set an object’s spacing relative to its left and right `BreakAlignments`.

For example, a `MultiMeasureRest` will ignore prefatory items at its bounds (i.e., clefs, key signatures and time signatures) using the following override:

```
\override MultiMeasureRest
  #'spacing-pair = #'(staff-bar . staff-bar)
```

spanner-id (string)

An identifier to distinguish concurrent spanners.

springs-and-rods (boolean)

Dummy variable for triggering spacing routines.

stacking-dir (direction)

Stack objects in which direction?

staff-affinity (direction)

The direction of the staff to use for spacing the current non-staff line. Choices are `UP`, `DOWN`, and `CENTER`. If `CENTER`, the non-staff line will be placed equidistant between the two nearest staves on either side, unless collisions or other spacing constraints prevent this. Setting `staff-affinity` for a staff causes it to be treated as a non-staff line. Setting `staff-affinity` to `#f` causes a non-staff line to be treated as a staff.

staff-padding (dimension, in staff space)

Maintain this much space between reference points and the staff. Its effect is to align objects of differing sizes (like the dynamics **p** and **f**) on their baselines.

staff-position (number)

Vertical position, measured in half staff spaces, counted from the middle line.

staff-space (dimension, in staff space)

Amount of space between staff lines, expressed in global `staff-space`.

staff-staff-spacing (list)

When applied to a staff-group’s `StaffGrouper` grob, this spacing alist controls the distance between consecutive staves within the staff-group. When applied to a staff’s `VerticalAxisGroup` grob, it controls the distance between the staff and the nearest staff below it in the same system, replacing any settings inherited from the `StaffGrouper` grob of the containing staff-group, if there is one. This property remains in effect even when non-staff lines appear between staves. The alist can contain the following keys:

- **basic-distance** – the vertical distance, measured in staff-spaces, between the reference points of the two items when no collisions would result, and no stretching or compressing is in effect.
- **minimum-distance** – the smallest allowable vertical distance, measured in staff-spaces, between the reference points of the two items, when compressing is in effect.
- **padding** – the minimum required amount of unobstructed vertical whitespace between the bounding boxes (or skylines) of the two items, measured in staff-spaces.
- **stretchability** – a unitless measure of the dimension’s relative propensity to stretch. If zero, the distance will not stretch (unless collisions would result).

staffgroup-staff-spacing (list)

The spacing alist controlling the distance between the last staff of the current staff-group and the staff just below it in the same system, even if one or more non-staff lines exist between the two staves. If the **staff-staff-spacing** property of the staff’s **VerticalAxisGroup** grob is set, that is used instead. See **staff-staff-spacing** for a description of the alist structure.

stem-attachment (pair of numbers)

An (x . y) pair where the stem attaches to the notehead.

stem-begin-position (number)

User override for the begin position of a stem.

stem-spacing-correction (number)

Optical correction amount for stems that are placed in tight configurations. For opposite directions, this amount is the correction for two normal sized stems that overlap completely.

stemlet-length (number)

How long should be a stem over a rest?

stencil (stencil)

The symbol to print.

stencils (list)

Multiple stencils, used as intermediate value.

strict-grace-spacing (boolean)

If set, main notes are spaced normally, then grace notes are put left of the musical columns for the main notes.

strict-note-spacing (boolean)

If set, unbroken columns with non-musical material (clefs, bar lines, etc.) are not spaced separately, but put before musical columns.

stroke-style (string)

Set to "grace" to turn stroke through flag on.

style (symbol)

This setting determines in what style a grob is typeset. Valid choices depend on the **stencil** callback reading this property.

text (markup)

Text markup. See [Section “Formatting text” dans *Manuel de notation*](#).

text-direction (direction)

This controls the ordering of the words. The default **RIGHT** is for roman text. Arabic or Hebrew should use **LEFT**.

thick-thickness (number)

Bar line thickness, measured in **line-thickness**.

thickness (number)

Line thickness, generally measured in **line-thickness**.

thin-kern (number)

The space after a hair-line in a bar line.

tie-configuration (list)

List of (**position** . **dir**) pairs, indicating the desired tie configuration, where *position* is the offset from the center of the staff in staff space and *dir* indicates the direction of the tie (1=>up, -1=>down, 0=>center). A non-pair entry in the list causes the corresponding tie to be formatted automatically.

to-barline (boolean)

If true, the spanner will stop at the bar line just before it would otherwise stop.

toward-stem-shift (number)

Amount by which scripts are shifted toward the stem if their direction coincides with the stem direction. 0.0 means keep the default position (centered on the note head), 1.0 means centered on the stem. Interpolated values are possible.

transparent (boolean)

This makes the grob invisible.

uniform-stretching (boolean)

If set, items stretch proportionally to their durations. This looks better in complex polyphonic patterns.

usable-duration-logs (list)

List of **duration-logs** that can be used in typesetting the grob.

use-skylines (boolean)

Should skylines be used for side positioning?

used (boolean)

If set, this spacing column is kept in the spacing problem.

vertical-skylines (pair of skylines)

Two skylines, one above and one below this grob.

when (moment)

Global time step associated with this column happen?

whiteout (boolean)

If true, the grob is printed over a white background to white-out underlying material, if the grob is visible. Usually #f by default.

width (dimension, in staff space)

The width of a grob measured in staff space.

word-space (dimension, in staff space)

Space to insert between words in texts.

X-extent (pair of numbers)

Hard coded extent in X direction.

X-offset (number)

The horizontal amount that this object is moved relative to its X-parent.

X-positions (pair of numbers)

Pair of X staff coordinates of a spanner in the form (*left* . *right*), where both *left* and *right* are in **staff-space** units of the current staff.

Y-extent (pair of numbers)

Hard coded extent in Y direction.

Y-offset (number)

The vertical amount that this object is moved relative to its Y-parent.

zigzag-length (dimension, in staff space)

The length of the lines of a zigzag, relative to **zigzag-width**. A value of 1 gives 60-degree zigzags.

zigzag-width (dimension, in staff space)

The width of one zigzag squiggle. This number is adjusted slightly so that the glissando line can be constructed from a whole number of squiggles.

A.18 Fonctions musicales prédéfinies

acciaccatura [music] - *music* (music)

Create an acciaccatura from the following music expression

accidentalStyle [music] - *context* [symbol] *style* (string)

Set accidental style to *style*, a string. If an optional *context* symbol is given, e.g. **#'Staff** or **#'Voice**, the settings are applied to that context. Otherwise, the context defaults to **'Staff'**, except for piano styles, which use **'GrandStaff'** as a context.

addChordShape [music] - *key-symbol* (symbol) *tuning* (pair) *shape-definition* (string or pair)

Add chord shape *shape-definition* to the *chord-shape-table* hash with the key (**cons** *key-symbol* *tuning*).

addInstrumentDefinition [void] - *name* (string) *lst* (list)

Create instrument *name* with properties *list*.

addQuote [void] - *name* (string) *music* (music)

Define *music* as a quotable music expression named *name*

afterGrace [music] - *main* (music) *grace* (music)

Create *grace* note(s) after a *main* music expression.

allowPageTurn [music]

Allow a page turn. May be used at toplevel (ie between scores or markups), or inside a score.

alterBroken [music] - *name* (string) *property* (any type) *arg* (list)

Override *property* for pieces of broken spanner *name* with values *arg*.

appendToTag [music] - *tag* (symbol) *more* (music) *music* (music)

Append *more* to the **elements** of all music expressions in *music* that are tagged with *tag*.

applyContext [music] - *proc* (procedure)

Modify context properties with Scheme procedure *proc*.

applyMusic [music] - *func* (procedure) *music* (music)

Apply procedure *func* to *music*.

applyOutput [music] - *ctx* (symbol) *proc* (procedure)

Apply function *proc* to every layout object in context *ctx*

- appoggiatura** [music] - *music* (music)
Create an appoggiatura from *music*
- assertBeamQuant** [music] - *l* (pair) *r* (pair)
Testing function: check whether the beam quants *l* and *r* are correct
- assertBeamSlope** [music] - *comp* (procedure)
Testing function: check whether the slope of the beam is the same as *comp*
- autochange** [music] - *music* (music)
Make voices that switch between staves automatically
- balloonGrobText** [music] - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)
Attach *text* to *grob-name* at offset *offset* (use like `\once`)
- balloonText** [music] - *offset* (pair of numbers) *text* (markup)
Attach *text* at *offset* (use like `\tweak`)
- bar** [music] - *type* (string)
Insert a bar line of type *type*
- barNumberCheck** [music] - *n* (integer)
Print a warning if the current bar number is not *n*.
- bendAfter** [post event] - *delta* (real number)
Create a fall or doit of pitch interval *delta*.
- bookOutputName** [void] - *newfilename* (string)
Direct output for the current book block to *newfilename*.
- bookOutputSuffix** [void] - *newsuffix* (string)
Set the output filename suffix for the current book block to *newsuffix*.
- breathe** [music]
Insert a breath mark.
- chordRepeats** [music] - *event-types* [list] *music* (music)
Walk through *music* putting the notes of the previous chord into repeat chords, as well as an optional list of *event-types* such as `#'(string-number-event)`.
- clef** [music] - *type* (string)
Set the current clef to *type*.
- compoundMeter** [music] - *args* (pair)
Create compound time signatures. The argument is a Scheme list of lists. Each list describes one fraction, with the last entry being the denominator, while the first entries describe the summands in the numerator. If the time signature consists of just one fraction, the list can be given directly, i.e. not as a list containing a single list. For example, a time signature of $(3+1)/8 + 2/4$ would be created as `\compoundMeter #'((3 1 8) (2 4))`, and a time signature of $(3+2)/8$ as `\compoundMeter #'((3 2 8))` or shorter `\compoundMeter #'(3 2 8)`.
- crossStaff** [music] - *notes* (music)
Create cross-staff stems
- cueClef** [music] - *type* (string)
Set the current cue clef to *type*.
- cueClefUnset** [music]
Unset the current cue clef.

- cueDuring** [music] - *what* (string) *dir* (direction) *main-music* (music)
Insert contents of quote *what* corresponding to *main-music*, in a CueVoice oriented by *dir*.
- cueDuringWithClef** [music] - *what* (string) *dir* (direction) *clef* (string) *main-music* (music)
Insert contents of quote *what* corresponding to *main-music*, in a CueVoice oriented by *dir*.
- deadNote** [music] - *note* (music)
Print *note* with a cross-shaped note head.
- defaultNoteHeads** [music]
Revert to the default note head style.
- displayLilyMusic** [music] - *music* (music)
Display the LilyPond input representation of *music* to the console.
- displayMusic** [music] - *music* (music)
Display the internal representation of *music* to the console.
- endSpanners** [music] - *music* (music)
Terminate the next spanner prematurely after exactly one note without the need of a specific end spanner.
- eventChords** [music] - *music* (music)
Compatibility function wrapping **EventChord** around isolated rhythmic events occurring since version 2.15.28, after expanding repeat chords ‘q’.
- featherDurations** [music] - *factor* (moment) *argument* (music)
Adjust durations of music in *argument* by rational *factor*.
- footnote** [music] - *mark* [markup] *offset* (pair of numbers) *grob-name* [symbol] *footnote* (markup) *music* [music]
Make the markup *footnote* a footnote on *music*. The footnote is marked with a markup *mark* moved by *offset* with respect to the marked music.
If *mark* is not given or specified as `\default`, it is replaced by an automatically generated sequence number. If a symbol *grob-name* is specified, then grobs of that type will be marked if they have *music* as their ultimate cause; by default all grobs having *music* as their *direct* cause will be marked, similar to the way `\tweak` works.
If *music* is given as `\default`, a footnote event affecting *all* grobs matching *grob-name* at a given time step is generated. This may be required for creating footnotes on time signatures, clefs, and other items not cooperating with `\tweak`.
Like with `\tweak`, if you use a footnote on a following post-event, the `\footnote` command itself needs to be attached to the preceding note or rest as a post-event with `-`.
- grace** [music] - *music* (music)
Insert *music* as grace notes.
- grobdescriptions** (any type) - *descriptions* (list)
Create a context modification from *descriptions*, a list in the format of **all-grob-descriptions**.
- harmonicByFret** [music] - *fret* (number) *music* (music)
Convert *music* into mixed harmonics; the resulting notes resemble harmonics played on a fretted instrument by touching the strings at *fret*.
- harmonicByRatio** [music] - *ratio* (number) *music* (music)
Convert *music* into mixed harmonics; the resulting notes resemble harmonics played on a fretted instrument by touching the strings at the point given through *ratio*.

- harmonicNote** [music] - *note* (music)
Print *note* with a diamond-shaped note head.
- harmonicsOn** [music]
Set the default note head style to a diamond-shaped style.
- inStaffSegno** [music]
Put the segno variant 'varsegno' at this position into the staff, compatible with the repeat command.
- instrumentSwitch** [music] - *name* (string)
Switch instrument to *name*, which must be predefined with `\addInstrumentDefinition`.
- inversion** [music] - *around* (pitch) *to* (pitch) *music* (music)
Invert *music* about *around* and transpose from *around* to *to*.
- keepWithTag** [music] - *tag* (symbol) *music* (music)
Include only elements of *music* that are tagged with *tag*.
- key** [music] - *tonic* [pitch] *pitch-alist* [list]
Set key to *tonic* and scale *pitch-alist*. If both are null, just generate `KeyChangeEvent`.
- killCues** [music] - *music* (music)
Remove cue notes from *music*.
- label** [music] - *label* (symbol)
Create *label* as a bookmarking label.
- language** [void] - *language* (string)
Set note names for language *language*.
- languageRestore** [void]
Restore a previously-saved pitchnames alist.
- languageSaveAndChange** [void] - *language* (string)
Store the previous pitchnames alist, and set a new one.
- makeClusters** [music] - *arg* (music)
Display chords in *arg* as clusters.
- makeDefaultStringTuning** [void] - *symbol* (symbol) *pitches* (list)
This defines a string tuning *symbol* via a list of *pitches*. The *symbol* also gets registered in `defaultStringTunings` for documentation purposes.
- mark** [music] - *label* [any type]
Make the music for the `\mark` command.
- modalInversion** [music] - *around* (pitch) *to* (pitch) *scale* (music) *music* (music)
Invert *music* about *around* using *scale* and transpose from *around* to *to*.
- modalTranspose** [music] - *from* (pitch) *to* (pitch) *scale* (music) *music* (music)
Transpose *music* from pitch *from* to pitch *to* using *scale*.
- musicMap** [music] - *proc* (procedure) *mus* (music)
Apply *proc* to *mus* and all of the music it contains.
- noPageBreak** [music]
Forbid a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.
- noPageTurn** [music]
Forbid a page turn. May be used at toplevel (i.e., between scores or markups), or inside a score.

octaveCheck [music] - *pitch* (pitch)
Octave check.

once [music] - *music* (music)
Set **once** to **#t** on all layout instruction events in *music*.

ottava [music] - *octave* (integer)
Set the octavation.

overrideProperty [music] - *name* (string) *property* (symbol) *value* (any type)
Set *property* to *value* in all grobs named *name*. The *name* argument is a string of the form "Context.GrobName" or "GrobName".

overrideTimeSignatureSettings [music] - *time-signature* (pair) *base-moment* (pair)
beat-structure (list) *beam-exceptions* (list)
Override **timeSignatureSettings** for time signatures of *time-signature* to have settings of *base-moment*, *beat-structure*, and *beam-exceptions*.

pageBreak [music]
Force a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.

pageTurn [music]
Force a page turn between two scores or top-level markups.

palmMute [music] - *note* (music)
Print *note* with a triangle-shaped note head.

palmMuteOn [music]
Set the default note head style to a triangle-shaped style.

parallelMusic [void] - *voice-ids* (list) *music* (music)
Define parallel music sequences, separated by '|' (bar check signs), and assign them to the identifiers provided in *voice-ids*.
voice-ids: a list of music identifiers (symbols containing only letters)
music: a music sequence, containing BarChecks as limiting expressions.
Example:

```
\parallelMusic #'(A B C) {
  c c | d d | e e |
  d d | e e | f f |
}
<==>
A = { c c | d d | }
B = { d d | e e | }
C = { e e | f f | }
```

parenthesize [music] - *arg* (music)
Tag *arg* to be parenthesized.

partcombine [music] - *part1* (music) *part2* (music)
Take the music in *part1* and *part2* and typeset so that they share a staff.

partcombineDown [music] - *part1* (music) *part2* (music)
Take the music in *part1* and *part2* and typeset so that they share a staff with stems directed downward.

partcombineForce [music] - *type* (symbol-or-boolean) *once* (boolean)
Override the part-combiner.

- partcombineUp** [music] - *part1* (music) *part2* (music)
Take the music in *part1* and *part2* and typeset so that they share a staff with stems directed upward.
- partial** [music] - *dur* (duration)
Make a partial measure.
- phrasingSlurDashPattern** [music] - *dash-fraction* (number) *dash-period* (number)
Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval for phrasing slurs.
- pitchedTrill** [music] - *main-note* (music) *secondary-note* (music)
Print a trill with *main-note* as the main note of the trill and print *secondary-note* as a stemless note head in parentheses.
- pointAndClickOff** [music]
Suppress generating extra code in final-format (e.g. pdf) files to point back to the lilypond source statement.
- pointAndClickOn** [music]
Enable generation of code in final-format (e.g. pdf) files to reference the originating lilypond source statement; this is helpful when developing a score but generates bigger final-format files.
- pointAndClickTypes** [void] - *types* (list or symbol)
Set a type or list of types (such as `#'note-event`) for which point-and-click info is generated.
- pushToTag** [music] - *tag* (symbol) *more* (music) *music* (music)
Add *more* to the front of `elements` of all music expressions in *music* that are tagged with *tag*.
- quoteDuring** [music] - *what* (string) *main-music* (music)
Indicate a section of music to be quoted. *what* indicates the name of the quoted voice, as specified in an `\addQuote` command. *main-music* is used to indicate the length of music to be quoted; usually contains spacers or multi-measure rests.
- relative** [music] - *pitch* [pitch] *music* (music)
Make *music* relative to *pitch* (default `c'`).
- removeWithTag** [music] - *tag* (symbol) *music* (music)
Remove elements of *music* that are tagged with *tag*.
- resetRelativeOctave** [music] - *pitch* (pitch)
Set the octave inside a `\relative` section.
- retrograde** [music] - *music* (music)
Return *music* in reverse order.
- revertTimeSignatureSettings** [music] - *time-signature* (pair)
Revert `timeSignatureSettings` for time signatures of *time-signature*.
- rightHandFinger** [post event] - *finger* (number or string)
Apply *finger* as a fingering indication.
- scaleDurations** [music] - *fraction* (fraction, as pair) *music* (music)
Multiply the duration of events in *music* by *fraction*.
- settingsFrom** (any type) - *ctx* [symbol] *music* (music)
Take the layout instruction events from *music*, optionally restricted to those applying to context type *ctx*, and return a context modification duplicating their effect.

- shape** [music] - *grob* (string) *offsets* (list)
Offset control-points of *grob* by *offsets*. The argument is a list of number pairs or list of such lists. Each element of a pair represents an offset to one of the coordinates of a control-point.
- shiftDurations** [music] - *dur* (integer) *dots* (integer) *arg* (music)
Change the duration of *arg* by adding *dur* to the **durlog** of *arg* and *dots* to the **dots** of *arg*.
- skip** [music] - *dur* (duration)
Skip forward by *dur*.
- slashedGrace** [music] - *music* (music)
Create slashed graces (slashes through stems, but no slur) from the following music expression
- slurDashPattern** [music] - *dash-fraction* (number) *dash-period* (number)
Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval for slurs.
- spacingTweaks** [music] - *parameters* (list)
Set the system stretch, by reading the 'system-stretch' property of the 'parameters' assoc list.
- storePredefinedDiagram** [music] - *fretboard-table* (hash table) *chord* (music) *tuning* (pair) *diagram-definition* (string or pair)
Add predefined fret diagram defined by *diagram-definition* for the chord pitches *chord* and the stringTuning *tuning*.
- stringTuning** (any type) - *chord* (music)
Convert *chord* to a string tuning. *chord* must be in absolute pitches and should have the highest string number (generally the lowest pitch) first.
- styledNoteHeads** [music] - *style* (symbol) *heads* (list or symbol) *music* (music)
Set *heads* in *music* to *style*.
- tabChordRepeats** [music] - *event-types* [list] *music* (music)
Walk through *music* putting the notes, fingerings and string numbers of the previous chord into repeat chords, as well as an optional list of *event-types* such as **#'(articulation-event)**.
- tabChordRepetition** [void]
Include the string and fingering information in a chord repetition. This function is deprecated; try using **abChordRepeats** instead.
- tag** [music] - *tag* (symbol) *arg* (music)
Add *tag* to the **tags** property of *arg*.
- tieDashPattern** [music] - *dash-fraction* (number) *dash-period* (number)
Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval for ties.
- time** [music] - *beat-structure* [number list] *fraction* (fraction, as pair)
Set *fraction* as time signature, with optional number list *beat-structure* before it.
- times** [music] - *fraction* (fraction, as pair) *music* (music)
Scale *music* in time by *fraction*.
- tocItem** [music] - *text* (markup)
Add a line to the table of content, using the **tocItemMarkup** paper variable markup

- transpose** [music] - *from* (pitch) *to* (pitch) *music* (music)
Transpose *music* from pitch *from* to pitch *to*.
- transposedCueDuring** [music] - *what* (string) *dir* (direction) *pitch* (pitch) *main-music* (music)
Insert notes from the part *what* into a voice called **cue**, using the transposition defined by *pitch*. This happens simultaneously with *main-music*, which is usually a rest. The argument *dir* determines whether the cue notes should be notated as a first or second voice.
- transposition** [music] - *pitch* (pitch)
Set instrument transposition
- tweak** [music] - *grob* [string] *prop* (symbol) *value* (any type) *music* (music)
Add a tweak to the following *music*. Layout objects created by *music* get their property *prop* set to *value*. If *grob* is specified, like with
`\tweak Accidental #'color #red cis'`
 an indirectly created grob ('Accidental' is caused by 'NoteHead') can be tweaked; otherwise only directly created grobs are affected.
- unfoldRepeats** [music] - *music* (music)
Force any `\repeat volta`, `\repeat tremolo` or `\repeat percent` commands in *music* to be interpreted as `\repeat unfold`.
- void** [void] - *arg* (any type)
Accept a scheme argument, return a void expression. Use this if you want to have a scheme expression evaluated because of its side-effects, but its value ignored.
- withMusicProperty** [music] - *sym* (symbol) *val* (any type) *music* (music)
Set *sym* to *val* in *music*.
- xNote** [music] - *note* (music)
Print *note* with a cross-shaped note head.
- xNotesOn** [music]
Set the default note head style to a cross-shaped style.

A.19 Identificateurs de modification de contexte

Les commandes suivantes permettent de modifier des contextes au sein d'un bloc `\layout` ou `\with`.

RemoveEmptyStaves

Remove staves which are considered to be empty according to the list of interfaces set by **keepAliveInterfaces**.

- Removes Section `'Axis_group_engraver'` dans *Référence des propriétés internes*.
- Removes Section `'Hara_kiri_engraver'` dans *Référence des propriétés internes*.
- Adds Section `'Hara_kiri_engraver'` dans *Référence des propriétés internes*.
- Sets grob property `remove-empty` in Section `'VerticalAxisGroup'` dans *Référence des propriétés internes* to `#t`.

A.20 Types de prédicats prédéfinis

R5RS primary predicates

Type predicate	Description
<code>boolean?</code>	boolean
<code>char?</code>	character
<code>number?</code>	number
<code>pair?</code>	pair
<code>port?</code>	port
<code>procedure?</code>	procedure
<code>string?</code>	string
<code>symbol?</code>	symbol
<code>vector?</code>	vector

R5RS secondary predicates

Type predicate	Description
<code>char-alphabetic?</code>	alphabetic character
<code>char-lower-case?</code>	lower-case character
<code>char-numeric?</code>	numeric character
<code>char-upper-case?</code>	upper-case character
<code>char-whitespace?</code>	whitespace character
<code>complex?</code>	complex number
<code>eof-object?</code>	end-of-file object
<code>even?</code>	even number
<code>exact?</code>	exact number
<code>inexact?</code>	inexact number
<code>input-port?</code>	input port
<code>integer?</code>	integer
<code>list?</code>	list (<i>use <code>cheap-list?</code> for faster processing</i>)
<code>negative?</code>	negative number
<code>null?</code>	null
<code>odd?</code>	odd number
<code>output-port?</code>	output port
<code>positive?</code>	positive number
<code>rational?</code>	rational number
<code>real?</code>	real number
<code>zero?</code>	zero

Guile predicates

Type predicate	Description
<code>hash-table?</code>	hash table

LilyPond scheme predicates

Type predicate	Description
<code>boolean-or-symbol?</code>	boolean or symbol
<code>cheap-list?</code>	list (<i>use this instead of <code>list?</code> for faster processing</i>)
<code>color?</code>	color
<code>fraction?</code>	fraction, as pair
<code>grob-list?</code>	list of grobs

index?	non-negative integer
list-or-symbol?	list or symbol
markup?	markup
markup-command-list?	markup command list
markup-list?	markup list
moment-pair?	pair of moment objects
number-list?	number list
number-or-grob?	number or grob
number-or-pair?	number or pair
number-or-string?	number or string
number-pair?	pair of numbers
rhythmic-location?	rhythmic location
scheme?	any type
string-or-pair?	string or pair
string-or-symbol?	string or symbol
void?	void

LilyPond exported predicates

Type predicate	Description
ly:book?	book
ly:box?	box
ly:context?	context
ly:context-def?	context definition
ly:context-mod?	context modification
ly:dimension?	dimension, in staff space
ly:dir?	direction
ly:dispatcher?	dispatcher
ly:duration?	duration
ly:event?	post event
ly:font-metric?	font metric
ly:grob?	graphical (layout) object
ly:grob-array?	array of grobs
ly:input-location?	input location
ly:item?	item
ly:iterator?	iterator
ly:lily-lexer?	lily-lexer
ly:lily-parser?	lily-parser
ly:listener?	listener
ly:moment?	moment
ly:music?	music
ly:music-function?	music function
ly:music-list?	list of music objects
ly:music-output?	music output
ly:otf-font?	OpenType font
ly:output-def?	output definition
ly:page-marker?	page marker
ly:pango-font?	pango font
ly:paper-book?	paper book
ly:paper-system?	paper-system Prob
ly:pitch?	pitch

<code>ly:prob?</code>	property object
<code>ly:score?</code>	score
<code>ly:simple-closure?</code>	simple closure
<code>ly:skyline?</code>	skyline
<code>ly:skyline-pair?</code>	pair of skylines
<code>ly:source-file?</code>	source file
<code>ly:spanner?</code>	spanner
<code>ly:spring?</code>	spring
<code>ly:stencil?</code>	stencil
<code>ly:stream-event?</code>	stream event
<code>ly:translator?</code>	translator
<code>ly:translator-group?</code>	translator group
<code>ly:unpure-pure-container?</code>	unpure/pure container

A.21 Fonctions Scheme

<code>ly:add-context-mod</code>	<i>contextmods</i> <i>modification</i>	[Fonction]
Adds the given context <i>modification</i> to the list <i>contextmods</i> of context modifications.		
<code>ly:add-file-name-alist</code>	<i>alist</i>	[Fonction]
Add mappings for error messages from <i>alist</i> .		
<code>ly:add-interface</code>	<i>iface</i> <i>desc</i> <i>props</i>	[Fonction]
Add a new grob interface. <i>iface</i> is the interface name, <i>desc</i> is the interface description, and <i>props</i> is the list of user-settable properties for the interface.		
<code>ly:add-listener</code>	<i>list</i> <i>disp</i> <i>cl</i>	[Fonction]
Add the listener <i>list</i> to the dispatcher <i>disp</i> . Whenever <i>disp</i> hears an event of class <i>cl</i> , it is forwarded to <i>list</i> .		
<code>ly:add-option</code>	<i>sym</i> <i>val</i> <i>description</i>	[Fonction]
Add a program option <i>sym</i> . <i>val</i> is the default value and <i>description</i> is a string description.		
<code>ly:all-grob-interfaces</code>		[Fonction]
Return the hash table with all grob interface descriptions.		
<code>ly:all-options</code>		[Fonction]
Get all option settings in an alist.		
<code>ly:all-stencil-expressions</code>		[Fonction]
Return all symbols recognized as stencil expressions.		
<code>ly:assoc-get</code>	<i>key</i> <i>alist</i> <i>default-value</i> <i>strict-checking</i>	[Fonction]
Return value if <i>key</i> in <i>alist</i> , else <i>default-value</i> (or <i>#f</i> if not specified). If <i>strict-checking</i> is set to <i>#t</i> and <i>key</i> is not in <i>alist</i> , a <i>programming-error</i> is output.		
<code>ly:axis-group-interface::add-element</code>	<i>grob</i> <i>grob-element</i>	[Fonction]
Set <i>grob</i> the parent of <i>grob-element</i> on all axes of <i>grob</i> .		
<code>ly:basic-progress</code>	<i>str</i> <i>rest</i>	[Fonction]
A Scheme callable function to issue a basic progress message <i>str</i> . The message is formatted with <i>format</i> and <i>rest</i> .		
<code>ly:beam-score-count</code>		[Fonction]
count number of beam scores.		

ly:book? <i>x</i>	[Fonction]
Is <i>x</i> a Book object?	
ly:book-add-bookpart! <i>book-smob book-part</i>	[Fonction]
Add <i>book-part</i> to <i>book-smob</i> book part list.	
ly:book-add-score! <i>book-smob score</i>	[Fonction]
Add <i>score</i> to <i>book-smob</i> score list.	
ly:book-book-parts <i>book</i>	[Fonction]
Return book parts in <i>book</i> .	
ly:book-header <i>book</i>	[Fonction]
Return header in <i>book</i> .	
ly:book-paper <i>book</i>	[Fonction]
Return paper in <i>book</i> .	
ly:book-process <i>book-smob default-paper default-layout output</i>	[Fonction]
Print book. <i>output</i> is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).	
ly:book-process-to-systems <i>book-smob default-paper default-layout output</i>	[Fonction]
Print book. <i>output</i> is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).	
ly:book-scores <i>book</i>	[Fonction]
Return scores in <i>book</i> .	
ly:book-set-header! <i>book module</i>	[Fonction]
Set the book header.	
ly:box? <i>x</i>	[Fonction]
Is <i>x</i> a Box object?	
ly:bp <i>num</i>	[Fonction]
<i>num</i> bigpoints (1/72th inch).	
ly:bracket <i>a iv t p</i>	[Fonction]
Make a bracket in direction <i>a</i> . The extent of the bracket is given by <i>iv</i> . The wings protrude by an amount of <i>p</i> , which may be negative. The thickness is given by <i>t</i> .	
ly:broadcast <i>disp ev</i>	[Fonction]
Send the stream event <i>ev</i> to the dispatcher <i>disp</i> .	
ly:camel-case->lisp-identifier <i>name-sym</i>	[Fonction]
Convert FooBar_Bla to foo-bar-bla style symbol.	
ly:chain-assoc-get <i>key achain default-value strict-checking</i>	[Fonction]
Return value for <i>key</i> from a list of alists <i>achain</i> . If no entry is found, return <i>default-value</i> or #f if <i>default-value</i> is not specified. With <i>strict-checking</i> set to #t , a programming_error is output in such cases.	
ly:check-expected-warnings	[Fonction]
Check whether all expected warnings have really been triggered.	

<code>ly:cm num</code> <i>num cm.</i>	[Fonction]
<code>ly:command-line-code</code> The Scheme code specified on command-line with ‘-e’.	[Fonction]
<code>ly:command-line-options</code> The Scheme options specified on command-line with ‘-d’.	[Fonction]
<code>ly:connect-dispatchers to from</code> Make the dispatcher <i>to</i> listen to events from <i>from</i> .	[Fonction]
<code>ly:context? x</code> Is <i>x</i> a Context object?	[Fonction]
<code>ly:context-current-moment context</code> Return the current moment of <i>context</i> .	[Fonction]
<code>ly:context-def? x</code> Is <i>x</i> a Context_def object?	[Fonction]
<code>ly:context-def-lookup def sym val</code> Return the value of <i>sym</i> in output definition <i>def</i> (e.g., <code>\paper</code>). If no value is found, return <i>val</i> or ‘()’ if <i>val</i> is undefined.	[Fonction]
<code>ly:context-def-modify def mod</code> Return the result of applying the context-mod <i>mod</i> to the context definition <i>def</i> . Does not change <i>def</i> .	[Fonction]
<code>ly:context-event-source context</code> Return event-source of context <i>context</i> .	[Fonction]
<code>ly:context-events-below context</code> Return a stream-distributor that distributes all events from <i>context</i> and all its subcontexts.	[Fonction]
<code>ly:context-find context name</code> Find a parent of <i>context</i> that has name or alias <i>name</i> . Return #f if not found.	[Fonction]
<code>ly:context-grob-definition context name</code> Return the definition of <i>name</i> (a symbol) within <i>context</i> as an alist.	[Fonction]
<code>ly:context-id context</code> Return the ID string of <i>context</i> , i.e., for <code>\context Voice = "one" ...</code> return the string one .	[Fonction]
<code>ly:context-mod? x</code> Is <i>x</i> a Context_mod object?	[Fonction]
<code>ly:context-mod-apply! context mod</code> Apply the context modification <i>mod</i> to <i>context</i> .	[Fonction]
<code>ly:context-name context</code> Return the name of <i>context</i> , i.e., for <code>\context Voice = "one" ...</code> return the symbol Voice .	[Fonction]
<code>ly:context-now context</code> Return now-moment of context <i>context</i> .	[Fonction]

ly:context-parent <i>context</i>	[Fonction]
Return the parent of <i>context</i> , #f if none.	
ly:context-property <i>context sym def</i>	[Fonction]
Return the value for property <i>sym</i> in <i>context</i> . If <i>def</i> is given, and property value is '(), return <i>def</i> .	
ly:context-property-where-defined <i>context name</i>	[Fonction]
Return the context above <i>context</i> where <i>name</i> is defined.	
ly:context-pushpop-property <i>context grob eltprop val</i>	[Fonction]
Do a single \override or \revert operation in <i>context</i> . The grob definition <i>grob</i> is extended with <i>eltprop</i> (if <i>val</i> is specified) or reverted (if unspecified).	
ly:context-set-property! <i>context name val</i>	[Fonction]
Set value of property <i>name</i> in context <i>context</i> to <i>val</i> .	
ly:context-unset-property <i>context name</i>	[Fonction]
Unset value of property <i>name</i> in context <i>context</i> .	
ly:debug <i>str rest</i>	[Fonction]
A Scheme callable function to issue a debug message <i>str</i> . The message is formatted with <i>format</i> and <i>rest</i> .	
ly:default-scale	[Fonction]
Get the global default scale.	
ly:dimension? <i>d</i>	[Fonction]
Return <i>d</i> as a number. Used to distinguish length variables from normal numbers.	
ly:dir? <i>s</i>	[Fonction]
Is <i>s</i> a direction? Valid directions are -1, 0, or 1, where -1 represents left or down, 1 represents right or up, and 0 represents a neutral direction.	
ly:dispatcher? <i>x</i>	[Fonction]
Is <i>x</i> a Dispatcher object?	
ly:duration? <i>x</i>	[Fonction]
Is <i>x</i> a Duration object?	
ly:duration<? <i>p1 p2</i>	[Fonction]
Is <i>p1</i> shorter than <i>p2</i> ?	
ly:duration->string <i>dur</i>	[Fonction]
Convert <i>dur</i> to a string.	
ly:duration-dot-count <i>dur</i>	[Fonction]
Extract the dot count from <i>dur</i> .	
ly:duration-factor <i>dur</i>	[Fonction]
Extract the compression factor from <i>dur</i> . Return it as a pair.	
ly:duration-length <i>dur</i>	[Fonction]
The length of the duration as a moment.	
ly:duration-log <i>dur</i>	[Fonction]
Extract the duration log from <i>dur</i> .	

ly:duration-scale <i>dur</i>	[Fonction]
Extract the compression factor from <i>dur</i> . Return it as a rational.	
ly:effective-prefix	[Fonction]
Return effective prefix.	
ly:encode-string-for-pdf <i>str</i>	[Fonction]
Encode the given string to either Latin1 (which is a subset of the PDFDocEncoding) or if that's not possible to full UTF-16BE with Byte-Order-Mark (BOM).	
ly:engraver-announce-end-grob <i>engraver grob cause</i>	[Fonction]
Announce the end of a grob (i.e., the end of a spanner) originating from given <i>engraver</i> instance, with <i>grob</i> being a grob. <i>cause</i> should either be another grob or a music event.	
ly:engraver-make-grob <i>engraver grob-name cause</i>	[Fonction]
Create a grob originating from given <i>engraver</i> instance, with given <i>grob-name</i> , a symbol. <i>cause</i> should either be another grob or a music event.	
ly:error <i>str rest</i>	[Fonction]
A Scheme callable function to issue the error <i>str</i> . The error is formatted with format and <i>rest</i> .	
ly:eval-simple-closure <i>delayed closure scm-start scm-end</i>	[Fonction]
Evaluate a simple <i>closure</i> with the given <i>delayed</i> argument. If <i>scm-start</i> and <i>scm-end</i> are defined, evaluate it purely with those start and end points.	
ly:event? <i>obj</i>	[Fonction]
Is <i>obj</i> a proper (non-rhythmic) event object?	
ly:event-deep-copy <i>m</i>	[Fonction]
Copy <i>m</i> and all sub expressions of <i>m</i> .	
ly:event-property <i>sev sym</i>	[Fonction]
Get the property <i>sym</i> of stream event <i>mus</i> . If <i>sym</i> is undefined, return '().	
ly:event-set-property! <i>ev sym val</i>	[Fonction]
Set property <i>sym</i> in event <i>ev</i> to <i>val</i> .	
ly:expand-environment <i>str</i>	[Fonction]
Expand \$VAR and \${VAR} in <i>str</i> .	
ly:expect-warning <i>str rest</i>	[Fonction]
A Scheme callable function to register a warning to be expected and subsequently suppressed. If the warning is not encountered, a warning about the missing warning will be shown. The message should be translated with (_ ...) and changing parameters given after the format string.	
ly:find-file <i>name</i>	[Fonction]
Return the absolute file name of <i>name</i> , or #f if not found.	
ly:font-config-add-directory <i>dir</i>	[Fonction]
Add directory <i>dir</i> to FontConfig.	
ly:font-config-add-font <i>font</i>	[Fonction]
Add font <i>font</i> to FontConfig.	
ly:font-config-display-fonts	[Fonction]
Dump a list of all fonts visible to FontConfig.	

- ly:font-config-get-font-file** *name* [Fonction]
Get the file for font *name*.
- ly:font-design-size** *font* [Fonction]
Given the font metric *font*, return the design size, relative to the current output-scale.
- ly:font-file-name** *font* [Fonction]
Given the font metric *font*, return the corresponding file name.
- ly:font-get-glyph** *font name* [Fonction]
Return a stencil from *font* for the glyph named *name*. If the glyph is not available, return an empty stencil.
Note that this command can only be used to access glyphs from fonts loaded with **ly:system-font-load**; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings **fetaMusic** and **fetaBraces**, respectively.
- ly:font-glyph-name-to-charcode** *font name* [Fonction]
Return the character code for glyph *name* in *font*.
Note that this command can only be used to access glyphs from fonts loaded with **ly:system-font-load**; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings **fetaMusic** and **fetaBraces**, respectively.
- ly:font-glyph-name-to-index** *font name* [Fonction]
Return the index for *name* in *font*.
Note that this command can only be used to access glyphs from fonts loaded with **ly:system-font-load**; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings **fetaMusic** and **fetaBraces**, respectively.
- ly:font-index-to-charcode** *font index* [Fonction]
Return the character code for *index* in *font*.
Note that this command can only be used to access glyphs from fonts loaded with **ly:system-font-load**; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings **fetaMusic** and **fetaBraces**, respectively.
- ly:font-magnification** *font* [Fonction]
Given the font metric *font*, return the magnification, relative to the current output-scale.
- ly:font-metric?** *x* [Fonction]
Is *x* a **Font_metric** object?
- ly:font-name** *font* [Fonction]
Given the font metric *font*, return the corresponding name.
- ly:font-sub-fonts** *font* [Fonction]
Given the font metric *font* of an OpenType font, return the names of the subfonts within *font*.
- ly:format** *str rest* [Fonction]
LilyPond specific format, supporting **~a** and **~[0-9]f**. Basic support for **~s** is also provided.
- ly:format-output** *context* [Fonction]
Given a global context in its final state, process it and return the **Music_output** object in its final state.
- ly:get-all-function-documentation** [Fonction]
Get a hash table with all LilyPond Scheme extension functions.

ly:get-all-translators	[Fonction]
Return a list of all translator objects that may be instantiated.	
ly:get-context-mods <i>contextmod</i>	[Fonction]
Returns the list of context modifications stored in <i>contextmod</i> .	
ly:get-option <i>var</i>	[Fonction]
Get a global option setting.	
ly:get-spacing-spec <i>from-scm to-scm</i>	[Fonction]
Return the spacing spec going between the two given grobs, <i>from-scm</i> and <i>to-scm</i> .	
ly:get-undead <i>undead</i>	[Fonction]
Get back object from <i>undead</i> .	
ly:gettext <i>original</i>	[Fonction]
A Scheme wrapper function for gettext .	
ly:grob? <i>x</i>	[Fonction]
Is <i>x</i> a Grob object?	
ly:grob-alist-chain <i>grob global</i>	[Fonction]
Get an alist chain for grob <i>grob</i> , with <i>global</i> as the global default. If unspecified, font-defaults from the layout block is taken.	
ly:grob-array? <i>x</i>	[Fonction]
Is <i>x</i> a Grob_array object?	
ly:grob-array->list <i>grob-arr</i>	[Fonction]
Return the elements of <i>grob-arr</i> as a Scheme list.	
ly:grob-array-length <i>grob-arr</i>	[Fonction]
Return the length of <i>grob-arr</i> .	
ly:grob-array-ref <i>grob-arr index</i>	[Fonction]
Retrieve the <i>index</i> th element of <i>grob-arr</i> .	
ly:grob-basic-properties <i>grob</i>	[Fonction]
Get the immutable properties of <i>grob</i> .	
ly:grob-chain-callback <i>grob proc sym</i>	[Fonction]
Find the callback that is stored as property <i>sym</i> of grob <i>grob</i> and chain <i>proc</i> to the head of this, meaning that it is called using <i>grob</i> and the previous callback's result.	
ly:grob-common-refpoint <i>grob other axis</i>	[Fonction]
Find the common refpoint of <i>grob</i> and <i>other</i> for <i>axis</i> .	
ly:grob-common-refpoint-of-array <i>grob others axis</i>	[Fonction]
Find the common refpoint of <i>grob</i> and <i>others</i> (a grob-array) for <i>axis</i> .	
ly:grob-default-font <i>grob</i>	[Fonction]
Return the default font for grob <i>grob</i> .	
ly:grob-extent <i>grob refp axis</i>	[Fonction]
Get the extent in <i>axis</i> direction of <i>grob</i> relative to the grob <i>refp</i> .	
ly:grob-interfaces <i>grob</i>	[Fonction]
Return the interfaces list of grob <i>grob</i> .	

- ly:grob-layout** *grob* [Fonction]
Get \layout definition from grob *grob*.
- ly:grob-object** *grob sym* [Fonction]
Return the value of a pointer in grob *grob* of property *sym*. It returns '()' (end-of-list) if *sym* is undefined in *grob*.
- ly:grob-original** *grob* [Fonction]
Return the unbroken original grob of *grob*.
- ly:grob-parent** *grob axis* [Fonction]
Get the parent of *grob*. *axis* is 0 for the X-axis, 1 for the Y-axis.
- ly:grob-pq<?** *a b* [Fonction]
Compare two grob priority queue entries. This is an internal function.
- ly:grob-properties** *grob* [Fonction]
Get the mutable properties of *grob*.
- ly:grob-property** *grob sym val* [Fonction]
Return the value for property *sym* of *grob*. If no value is found, return *val* or '()' if *val* is not specified.
- ly:grob-property-data** *grob sym* [Fonction]
Return the value for property *sym* of *grob*, but do not process callbacks.
- ly:grob-pure-height** *grob refp beg end val* [Fonction]
Return the pure height of *grob* given retpoint *refp*. If no value is found, return *val* or '()' if *val* is not specified.
- ly:grob-pure-property** *grob sym beg end val* [Fonction]
Return the pure value for property *sym* of *grob*. If no value is found, return *val* or '()' if *val* is not specified.
- ly:grob-relative-coordinate** *grob refp axis* [Fonction]
Get the coordinate in *axis* direction of *grob* relative to the grob *refp*.
- ly:grob-robust-relative-extent** *grob refp axis* [Fonction]
Get the extent in *axis* direction of *grob* relative to the grob *refp*, or (0,0) if empty.
- ly:grob-script-priority-less** *a b* [Fonction]
Compare two grobs by script priority. For internal use.
- ly:grob-set-nested-property!** *grob symlist val* [Fonction]
Set nested property *symlist* in grob *grob* to value *val*.
- ly:grob-set-object!** *grob sym val* [Fonction]
Set *sym* in grob *grob* to value *val*.
- ly:grob-set-parent!** *grob axis parent-grob* [Fonction]
Set *parent-grob* the parent of grob *grob* in axis *axis*.
- ly:grob-set-property!** *grob sym val* [Fonction]
Set *sym* in grob *grob* to value *val*.
- ly:grob-staff-position** *sg* [Fonction]
Return the Y-position of *sg* relative to the staff.

ly:grob-suicide! <i>grob</i>	[Fonction]
Kill <i>grob</i> .	
ly:grob-system <i>grob</i>	[Fonction]
Return the system grob of <i>grob</i> .	
ly:grob-translate-axis! <i>grob d a</i>	[Fonction]
Translate <i>grob</i> on axis <i>a</i> over distance <i>d</i> .	
ly:grob-vertical <? <i>a b</i>	[Fonction]
Does <i>a</i> lie above <i>b</i> on the page?	
ly:gulp-file <i>name size</i>	[Fonction]
Read <i>size</i> characters from the file <i>name</i> , and return its contents in a string. If <i>size</i> is undefined, the entire file is read. The file is looked up using the search path.	
ly:hash-table-keys <i>tab</i>	[Fonction]
Return a list of keys in <i>tab</i> .	
ly:inch <i>num</i>	[Fonction]
<i>num</i> inches.	
ly:input-both-locations <i>sip</i>	[Fonction]
Return input location in <i>sip</i> as (file-name first-line first-column last-line last-column).	
ly:input-file-line-char-column <i>sip</i>	[Fonction]
Return input location in <i>sip</i> as (file-name line char column).	
ly:input-location? <i>x</i>	[Fonction]
Is <i>x</i> an input-location?	
ly:input-message <i>sip msg rest</i>	[Fonction]
Print <i>msg</i> as a GNU compliant error message, pointing to the location in <i>sip</i> . <i>msg</i> is interpreted similar to format 's argument, using <i>rest</i> .	
ly:input-warning <i>sip msg rest</i>	[Fonction]
Print <i>msg</i> as a GNU compliant warning message, pointing to the location in <i>sip</i> . <i>msg</i> is interpreted similar to format 's argument, using <i>rest</i> .	
ly:interpret-music-expression <i>mus ctx</i>	[Fonction]
Interpret the music expression <i>mus</i> in the global context <i>ctx</i> . The context is returned in its final state.	
ly:interpret-stencil-expression <i>expr func arg1 offset</i>	[Fonction]
Parse <i>expr</i> , feed bits to <i>func</i> with first arg <i>arg1</i> having offset <i>offset</i> .	
ly:intlog2 <i>d</i>	[Fonction]
The 2-logarithm of 1/ <i>d</i> .	
ly:item? <i>g</i>	[Fonction]
Is <i>g</i> an Item object?	
ly:item-break-dir <i>it</i>	[Fonction]
The break status direction of item <i>it</i> . -1 means end of line, 0 unbroken, and 1 beginning of line.	

ly:iterator? <i>x</i>	[Fonction]
Is <i>x</i> a <code>Music_iterator</code> object?	
ly:lexer-keywords <i>lexer</i>	[Fonction]
Return a list of (KEY . CODE) pairs, signifying the LilyPond reserved words list.	
ly:lily-lexer? <i>x</i>	[Fonction]
Is <i>x</i> a <code>Lily_lexer</code> object?	
ly:lily-parser? <i>x</i>	[Fonction]
Is <i>x</i> a <code>Lily_parser</code> object?	
ly:listened-event-class? <i>disp cl</i>	[Fonction]
Does <i>disp</i> listen to any event type in the list <i>cl</i> ?	
ly:listened-event-types <i>disp</i>	[Fonction]
Return a list of all event types that <i>disp</i> listens to.	
ly:listener? <i>x</i>	[Fonction]
Is <i>x</i> a <code>Listener</code> object?	
ly:make-book <i>paper header scores</i>	[Fonction]
Make a <code>\book</code> of <i>paper</i> and <i>header</i> (which may be <code>#f</code> as well) containing <code>\scores</code> .	
ly:make-book-part <i>scores</i>	[Fonction]
Make a <code>\bookpart</code> containing <code>\scores</code> .	
ly:make-context-mod <i>mod-list</i>	[Fonction]
Creates a context modification, optionally initialized via the list of modifications <i>mod-list</i> .	
ly:make-dispatcher	[Fonction]
Return a newly created dispatcher.	
ly:make-duration <i>length dotcount num den</i>	[Fonction]
<i>length</i> is the negative logarithm (base 2) of the duration: 1 is a half note, 2 is a quarter note, 3 is an eighth note, etc. The number of dots after the note is given by the optional argument <i>dotcount</i> .	
The duration factor is optionally given by integers <i>num</i> and <i>den</i> , alternatively by a single rational number.	
A duration is a musical duration, i.e., a length of time described by a power of two (whole, half, quarter, etc.) and a number of augmentation dots.	
ly:make-event-class <i>context type</i>	[Fonction]
Make an event class (a list of types) from the given <i>type</i> within the global context containing <i>context</i> .	
ly:make-global-context <i>output-def</i>	[Fonction]
Set up a global interpretation context, using the output block <i>output-def</i> . The context is returned.	
ly:make-global-translator <i>global</i>	[Fonction]
Create a translator group and connect it to the global context <i>global</i> . The translator group is returned.	
ly:make-listener <i>callback</i>	[Fonction]
Create a listener. Any time the listener hears an object, it will call <i>callback</i> with that object. <i>callback</i> should take exactly one argument.	

- ly:make-moment** *m g gn gd* [Fonction]
 Create the moment with rational main timing *m*, and optional grace timing *g*.
 A *moment* is a point in musical time. It consists of a pair of rationals (*m*, *g*), where *m* is the timing for the main notes, and *g* the timing for grace notes. In absence of grace notes, *g* is zero.
 For compatibility reasons, it is possible to write two numbers specifying numerator and denominator instead of the rationals. These forms cannot be mixed, and the two-argument form is disambiguated by the sign of the second argument: if it is positive, it can only be a denominator and not a grace timing.
- ly:make-music** *props* [Fonction]
 Make a C++ **Music** object and initialize it with *props*.
 This function is for internal use and is only called by **make-music**, which is the preferred interface for creating music objects.
- ly:make-music-function** *signature func* [Fonction]
 Make a function to process music, to be used for the parser. *func* is the function, and *signature* describes its arguments. *signature*'s cdr is a list containing either **ly:music?** predicates or other type predicates. Its car is the syntax function to call.
- ly:make-music-relative!** *music pitch* [Fonction]
 Make *music* relative to *pitch*, return final pitch.
- ly:make-output-def** [Fonction]
 Make an output definition.
- ly:make-page-label-marker** *label* [Fonction]
 Return page marker with label *label*.
- ly:make-page-permission-marker** *symbol permission* [Fonction]
 Return page marker with page breaking and turning permissions.
- ly:make-pango-description-string** *chain size* [Fonction]
 Make a PangoFontDescription string for the property alist *chain* at size *size*.
- ly:make-paper-outputter** *port format* [Fonction]
 Create an outputter that evaluates within *output-format*, writing to *port*.
- ly:make-pitch** *octave note alter* [Fonction]
octave is specified by an integer, zero for the octave containing middle C. *note* is a number indexing the global default scale, with 0 corresponding to pitch C and 6 usually corresponding to pitch B. Optional *alter* is a rational number of 200-cent whole tones for alteration.
- ly:make-prob** *type init rest* [Fonction]
 Create a Prob object.
- ly:make-scale** *steps* [Fonction]
 Create a scale. The argument is a vector of rational numbers, each of which represents the number of 200 cent tones of a pitch above the tonic.
- ly:make-score** *music* [Fonction]
 Return score with *music* encapsulated in it.
- ly:make-simple-closure** *expr* [Fonction]
 Make a simple closure. *expr* should be form of (*func a1 a2 ...*), and will be invoked as (*func delayed-arg a1 a2 ...*).

- ly:make-spring** *ideal min-dist* [Fonction]
 Make a spring. *ideal* is the ideal distance of the spring, and *min-dist* is the minimum distance.
- ly:make-stencil** *expr xext yext* [Fonction]
 Stencils are device independent output expressions. They carry two pieces of information:
1. A specification of how to print this object. This specification is processed by the output backends, for example ‘*scm/output-ps.scm*’.
 2. The vertical and horizontal extents of the object, given as pairs. If an extent is unspecified (or if you use (1000 . -1000) as its value), it is taken to be empty.
- ly:make-stream-event** *cl proplist* [Fonction]
 Create a stream event of class *cl* with the given mutable property list.
- ly:make-undead** *object* [Fonction]
 This packages *object* in a manner that keeps it from triggering "Parsed object should be dead" messages.
- ly:make-unpure-pure-container** *unpure pure* [Fonction]
 Make an unpure-pure container. *unpure* should be an unpure expression, and *pure* should be a pure expression. If *pure* is omitted, the value of *unpure* will be used twice.
- ly:message** *str rest* [Fonction]
 A Scheme callable function to issue the message *str*. The message is formatted with **format** and *rest*.
- ly:minimal-breaking** *pb* [Fonction]
 Break (pages and lines) the **Paper_book** object *pb* without looking for optimal spacing: stack as many lines on a page before moving to the next one.
- ly:mm** *num* [Fonction]
num mm.
- ly:module->alist** *mod* [Fonction]
 Dump the contents of module *mod* as an alist.
- ly:module-copy** *dest src* [Fonction]
 Copy all bindings from module *src* into *dest*.
- ly:modules-lookup** *modules sym def* [Fonction]
 Look up *sym* in the list *modules*, returning the first occurrence. If not found, return *def* or **#f** if *def* isn't specified.
- ly:moment?** *x* [Fonction]
 Is *x* a **Moment** object?
- ly:moment<?** *a b* [Fonction]
 Compare two moments.
- ly:moment-add** *a b* [Fonction]
 Add two moments.
- ly:moment-div** *a b* [Fonction]
 Divide two moments.
- ly:moment-grace** *mom* [Fonction]
 Extract grace timing as a rational number from *mom*.

<code>ly:moment-grace-denominator</code> <i>mom</i>	[Fonction]
Extract denominator from grace timing.	
<code>ly:moment-grace-numerator</code> <i>mom</i>	[Fonction]
Extract numerator from grace timing.	
<code>ly:moment-main</code> <i>mom</i>	[Fonction]
Extract main timing as a rational number from <i>mom</i> .	
<code>ly:moment-main-denominator</code> <i>mom</i>	[Fonction]
Extract denominator from main timing.	
<code>ly:moment-main-numerator</code> <i>mom</i>	[Fonction]
Extract numerator from main timing.	
<code>ly:moment-mod</code> <i>a b</i>	[Fonction]
Modulo of two moments.	
<code>ly:moment-mul</code> <i>a b</i>	[Fonction]
Multiply two moments.	
<code>ly:moment-sub</code> <i>a b</i>	[Fonction]
Subtract two moments.	
<code>ly:music?</code> <i>obj</i>	[Fonction]
Is <i>obj</i> a music object?	
<code>ly:music-compress</code> <i>m factor</i>	[Fonction]
Compress music object <i>m</i> by moment <i>factor</i> .	
<code>ly:music-deep-copy</code> <i>m</i>	[Fonction]
Copy <i>m</i> and all sub expressions of <i>m</i> .	
<code>ly:music-duration-compress</code> <i>mus fact</i>	[Fonction]
Compress <i>mus</i> by factor <i>fact</i> , which is a Moment .	
<code>ly:music-duration-length</code> <i>mus</i>	[Fonction]
Extract the duration field from <i>mus</i> and return the length.	
<code>ly:music-function?</code> <i>x</i>	[Fonction]
Is <i>x</i> a music-function?	
<code>ly:music-function-extract</code> <i>x</i>	[Fonction]
Return the Scheme function inside <i>x</i> .	
<code>ly:music-function-signature</code> <i>x</i>	[Fonction]
Return the function signature inside <i>x</i> .	
<code>ly:music-length</code> <i>mus</i>	[Fonction]
Get the length of music expression <i>mus</i> and return it as a Moment object.	
<code>ly:music-list?</code> <i>lst</i>	[Fonction]
Is <i>lst</i> a list of music objects?	
<code>ly:music-mutable-properties</code> <i>mus</i>	[Fonction]
Return an alist containing the mutable properties of <i>mus</i> . The immutable properties are not available, since they are constant and initialized by the <code>make-music</code> function.	

<code>ly:music-output? x</code>	[Fonction]
Is <i>x</i> a <code>Music_output</code> object?	
<code>ly:music-property mus sym val</code>	[Fonction]
Return the value for property <i>sym</i> of music expression <i>mus</i> . If no value is found, return <i>val</i> or '()' if <i>val</i> is not specified.	
<code>ly:music-set-property! mus sym val</code>	[Fonction]
Set property <i>sym</i> in music expression <i>mus</i> to <i>val</i> .	
<code>ly:music-transpose m p</code>	[Fonction]
Transpose <i>m</i> such that central C is mapped to <i>p</i> . Return <i>m</i> .	
<code>ly:note-column-accidentals note-column</code>	[Fonction]
Return the <code>AccidentalPlacement</code> grob from <i>note-column</i> if any, or <code>SCM_EOL</code> otherwise.	
<code>ly:note-column-dot-column note-column</code>	[Fonction]
Return the <code>DotColumn</code> grob from <i>note-column</i> if any, or <code>SCM_EOL</code> otherwise.	
<code>ly:note-head::stem-attachment font-metric glyph-name</code>	[Fonction]
Get attachment in <i>font-metric</i> for attaching a stem to notehead <i>glyph-name</i> .	
<code>ly:number->string s</code>	[Fonction]
Convert <i>s</i> to a string without generating many decimals.	
<code>ly:one-line-breaking pb</code>	[Fonction]
Put each score on a single line, and put each line on its own page. The paper-width setting will be modified so that every page will be wider than the widest line.	
<code>ly:optimal-breaking pb</code>	[Fonction]
Optimally break (pages and lines) the <code>Paper_book</code> object <i>pb</i> to minimize badness in both vertical and horizontal spacing.	
<code>ly:option-usage port</code>	[Fonction]
Print <code>ly:set-option</code> usage. Optional <i>port</i> argument for the destination defaults to current output port.	
<code>ly:otf->cff otf-file-name</code>	[Fonction]
Convert the contents of an OTF file to a CFF file, returning it as a string.	
<code>ly:otf-font? font</code>	[Fonction]
Is <i>font</i> an OpenType font?	
<code>ly:otf-font-glyph-info font glyph</code>	[Fonction]
Given the font metric <i>font</i> of an OpenType font, return the information about named glyph <i>glyph</i> (a string).	
<code>ly:otf-font-table-data font tag</code>	[Fonction]
Extract a table <i>tag</i> from <i>font</i> . Return empty string for non-existent <i>tag</i> .	
<code>ly:otf-glyph-count font</code>	[Fonction]
Return the number of glyphs in <i>font</i> .	
<code>ly:otf-glyph-list font</code>	[Fonction]
Return a list of glyph names for <i>font</i> .	
<code>ly:output-def? def</code>	[Fonction]
Is <i>def</i> an output definition?	

ly:output-def-clone <i>def</i>	[Fonction]
Clone output definition <i>def</i> .	
ly:output-def-lookup <i>def sym val</i>	[Fonction]
Return the value of <i>sym</i> in output definition <i>def</i> (e.g., <code>\paper</code>). If no value is found, return <i>val</i> or <code>'()</code> if <i>val</i> is undefined.	
ly:output-def-parent <i>def</i>	[Fonction]
Return the parent output definition of <i>def</i> .	
ly:output-def-scope <i>def</i>	[Fonction]
Return the variable scope inside <i>def</i> .	
ly:output-def-set-variable! <i>def sym val</i>	[Fonction]
Set an output definition <i>def</i> variable <i>sym</i> to <i>val</i> .	
ly:output-description <i>output-def</i>	[Fonction]
Return the description of translators in <i>output-def</i> .	
ly:output-find-context-def <i>output-def context-name</i>	[Fonction]
Return an alist of all context defs (matching <i>context-name</i> if given) in <i>output-def</i> .	
ly:output-formats	[Fonction]
Formats passed to <code>--format</code> as a list of strings, used for the output.	
ly:outputter-close <i>outputter</i>	[Fonction]
Close port of <i>outputter</i> .	
ly:outputter-dump-stencil <i>outputter stencil</i>	[Fonction]
Dump stencil <i>expr</i> onto <i>outputter</i> .	
ly:outputter-dump-string <i>outputter str</i>	[Fonction]
Dump <i>str</i> onto <i>outputter</i> .	
ly:outputter-module <i>outputter</i>	[Fonction]
Return output module of <i>outputter</i> .	
ly:outputter-output-scheme <i>outputter expr</i>	[Fonction]
Eval <i>expr</i> in module of <i>outputter</i> .	
ly:outputter-port <i>outputter</i>	[Fonction]
Return output port for <i>outputter</i> .	
ly:page-marker? <i>x</i>	[Fonction]
Is <i>x</i> a <code>Page_marker</code> object?	
ly:page-turn-breaking <i>pb</i>	[Fonction]
Optimally break (pages and lines) the <code>Paper_book</code> object <i>pb</i> such that page turns only happen in specified places, returning its pages.	
ly:pango-font? <i>f</i>	[Fonction]
Is <i>f</i> a pango font?	
ly:pango-font-physical-fonts <i>f</i>	[Fonction]
Return alist of (<code>ps-name file-name font-index</code>) lists for Pango font <i>f</i> .	
ly:paper-book? <i>x</i>	[Fonction]
Is <i>x</i> a <code>Paper_book</code> object?	

ly:paper-book-header <i>pb</i>	[Fonction]
Return the header definition (<code>\header</code>) in <code>Paper_book</code> object <i>pb</i> .	
ly:paper-book-pages <i>pb</i>	[Fonction]
Return pages in <code>Paper_book</code> object <i>pb</i> .	
ly:paper-book-paper <i>pb</i>	[Fonction]
Return the paper output definition (<code>\paper</code>) in <code>Paper_book</code> object <i>pb</i> .	
ly:paper-book-performances <i>pb</i>	[Fonction]
Return performances in <code>Paper_book</code> object <i>pb</i> .	
ly:paper-book-scopes <i>pb</i>	[Fonction]
Return scopes in <code>Paper_book</code> object <i>pb</i> .	
ly:paper-book-systems <i>pb</i>	[Fonction]
Return systems in <code>Paper_book</code> object <i>pb</i> .	
ly:paper-fonts <i>def</i>	[Fonction]
Return a list containing the fonts from output definition <i>def</i> (e.g., <code>\paper</code>).	
ly:paper-get-font <i>def chain</i>	[Fonction]
Find a font metric in output definition <i>def</i> satisfying the font-qualifiers in alist chain <i>chain</i> , and return it. (An alist chain is a list of alists, containing grob properties.)	
ly:paper-get-number <i>def sym</i>	[Fonction]
Return the value of variable <i>sym</i> in output definition <i>def</i> as a double.	
ly:paper-outputscales <i>def</i>	[Fonction]
Return the output-scale for output definition <i>def</i> .	
ly:paper-score-paper-systems <i>paper-score</i>	[Fonction]
Return vector of <code>paper_system</code> objects from <i>paper-score</i> .	
ly:paper-system? <i>obj</i>	[Fonction]
Is <i>obj</i> a C++ Prob object of type <code>paper-system</code> ?	
ly:paper-system-minimum-distance <i>sys1 sys2</i>	[Fonction]
Measure the minimum distance between these two paper-systems, using their stored skylines if possible and falling back to their extents otherwise.	
ly:parse-file <i>name</i>	[Fonction]
Parse a single .ly file. Upon failure, throw <code>ly-file-failed</code> key.	
ly:parse-string-expression <i>parser-smob ly-code filename line</i>	[Fonction]
Parse the string <i>ly-code</i> with <i>parser-smob</i> . Return the contained music expression. <i>filename</i> and <i>line</i> are optional source indicators.	
ly:parsed-undead-list!	[Fonction]
Return the list of objects that have been found live that should have been dead, and clear that list.	
ly:parser-clear-error <i>parser</i>	[Fonction]
Clear the error flag for the parser.	
ly:parser-clone <i>parser-smob closures</i>	[Fonction]
Return a clone of <i>parser-smob</i> . An association list of port positions to closures can be specified in <i>closures</i> in order to have <code>\$</code> and <code>#</code> interpreted in their original lexical environment.	

ly:parser-define! <i>parser-smob symbol val</i>	[Fonction]
Bind <i>symbol</i> to <i>val</i> in <i>parser-smob</i> 's module.	
ly:parser-error <i>parser msg input</i>	[Fonction]
Display an error message and make the parser fail.	
ly:parser-has-error? <i>parser</i>	[Fonction]
Does <i>parser</i> have an error flag?	
ly:parser-include-string <i>parser-smob ly-code</i>	[Fonction]
Include the string <i>ly-code</i> into the input stream for <i>parser-smob</i> . Can only be used in immediate Scheme expressions (\$ instead of #).	
ly:parser-lexer <i>parser-smob</i>	[Fonction]
Return the lexer for <i>parser-smob</i> .	
ly:parser-lookup <i>parser-smob symbol</i>	[Fonction]
Look up <i>symbol</i> in <i>parser-smob</i> 's module. Return '() if not defined.	
ly:parser-output-name <i>parser</i>	[Fonction]
Return the base name of the output file.	
ly:parser-parse-string <i>parser-smob ly-code</i>	[Fonction]
Parse the string <i>ly-code</i> with <i>parser-smob</i> . Upon failure, throw ly-file-failed key.	
ly:parser-set-note-names <i>parser names</i>	[Fonction]
Replace current note names in <i>parser</i> . <i>names</i> is an alist of symbols. This only has effect if the current mode is notes.	
ly:performance-write <i>performance filename</i>	[Fonction]
Write <i>performance</i> to <i>filename</i> .	
ly:pfb->pfa <i>pfb-file-name</i>	[Fonction]
Convert the contents of a Type 1 font in PFB format to PFA format.	
ly:pitch? <i>x</i>	[Fonction]
Is <i>x</i> a Pitch object?	
ly:pitch<? <i>p1 p2</i>	[Fonction]
Is <i>p1</i> lexicographically smaller than <i>p2</i> ?	
ly:pitch-alteration <i>pp</i>	[Fonction]
Extract the alteration from pitch <i>pp</i> .	
ly:pitch-diff <i>pitch root</i>	[Fonction]
Return pitch <i>delta</i> such that <i>pitch</i> transposed by <i>delta</i> equals <i>root</i> .	
ly:pitch-negate <i>p</i>	[Fonction]
Negate <i>p</i> .	
ly:pitch-notename <i>pp</i>	[Fonction]
Extract the note name from pitch <i>pp</i> .	
ly:pitch-octave <i>pp</i>	[Fonction]
Extract the octave from pitch <i>pp</i> .	
ly:pitch-quartertones <i>pp</i>	[Fonction]
Calculate the number of quarter tones of <i>pp</i> from middle C.	

ly:pitch-semitones <i>pp</i>	[Fonction]
Calculate the number of semitones of <i>pp</i> from middle C.	
ly:pitch-steps <i>p</i>	[Fonction]
Number of steps counted from middle C of the pitch <i>p</i> .	
ly:pitch-tones <i>pp</i>	[Fonction]
Calculate the number of tones of <i>pp</i> from middle C as a rational number.	
ly:pitch-transpose <i>p delta</i>	[Fonction]
Transpose <i>p</i> by the amount <i>delta</i> , where <i>delta</i> is relative to middle C.	
ly:pointer-group-interface::add-grob <i>grob sym grob-element</i>	[Fonction]
Add <i>grob-element</i> to <i>grob</i> 's <i>sym</i> grob array.	
ly:position-on-line? <i>sg spos</i>	[Fonction]
Return whether <i>spos</i> is on a line of the staff associated with the grob <i>sg</i> (even on an extender line).	
ly:prob? <i>x</i>	[Fonction]
Is <i>x</i> a Prob object?	
ly:prob-immutable-properties <i>prob</i>	[Fonction]
Retrieve an alist of immutable properties.	
ly:prob-mutable-properties <i>prob</i>	[Fonction]
Retrieve an alist of mutable properties.	
ly:prob-property <i>prob sym val</i>	[Fonction]
Return the value for property <i>sym</i> of Prob object <i>prob</i> . If no value is found, return <i>val</i> or '()' if <i>val</i> is not specified.	
ly:prob-property? <i>obj sym</i>	[Fonction]
Is boolean prop <i>sym</i> of <i>sym</i> set?	
ly:prob-set-property! <i>obj sym value</i>	[Fonction]
Set property <i>sym</i> of <i>obj</i> to <i>value</i> .	
ly:prob-type? <i>obj type</i>	[Fonction]
Is <i>obj</i> the specified prob-type?	
ly:programming-error <i>str rest</i>	[Fonction]
A Scheme callable function to issue the internal warning <i>str</i> . The message is formatted with format and <i>rest</i> .	
ly:progress <i>str rest</i>	[Fonction]
A Scheme callable function to print progress <i>str</i> . The message is formatted with format and <i>rest</i> .	
ly:property-lookup-stats <i>sym</i>	[Fonction]
Return hash table with a property access corresponding to <i>sym</i> . Choices are prob , grob , and context .	
ly:protects	[Fonction]
Return hash of protected objects.	
ly:pt <i>num</i>	[Fonction]
<i>num</i> printer points.	

ly:register-stencil-expression <i>symbol</i>	[Fonction]
Add <i>symbol</i> as head of a stencil expression.	
ly:relative-group-extent <i>elements common axis</i>	[Fonction]
Determine the extent of <i>elements</i> relative to <i>common</i> in the <i>axis</i> direction.	
ly:reset-all-fonts	[Fonction]
Forget all about previously loaded fonts.	
ly:round-filled-box <i>xext yext blot</i>	[Fonction]
Make a Stencil object that prints a black box of dimensions <i>xext</i> , <i>yext</i> and roundness <i>blot</i> .	
ly:round-filled-polygon <i>points blot</i>	[Fonction]
Make a Stencil object that prints a black polygon with corners at the points defined by <i>points</i> (list of coordinate pairs) and roundness <i>blot</i> .	
ly:run-translator <i>mus output-def</i>	[Fonction]
Process <i>mus</i> according to <i>output-def</i> . An interpretation context is set up, and <i>mus</i> is interpreted with it. The context is returned in its final state.	
Optionally, this routine takes an object-key to uniquely identify the score block containing it.	
ly:score? <i>x</i>	[Fonction]
Is <i>x</i> a Score object?	
ly:score-add-output-def! <i>score def</i>	[Fonction]
Add an output definition <i>def</i> to <i>score</i> .	
ly:score-embedded-format <i>score layout</i>	[Fonction]
Run <i>score</i> through <i>layout</i> (an output definition) scaled to correct output-scale already, returning a list of layout-lines.	
ly:score-error? <i>score</i>	[Fonction]
Was there an error in the score?	
ly:score-header <i>score</i>	[Fonction]
Return score header.	
ly:score-music <i>score</i>	[Fonction]
Return score music.	
ly:score-output-defs <i>score</i>	[Fonction]
All output definitions in a score.	
ly:score-set-header! <i>score module</i>	[Fonction]
Set the score header.	
ly:set-default-scale <i>scale</i>	[Fonction]
Set the global default scale. This determines the tuning of pitches with no accidentals or key signatures. The first pitch is C. Alterations are calculated relative to this scale. The number of pitches in this scale determines the number of scale steps that make up an octave. Usually the 7-note major scale.	
ly:set-grob-modification-callback <i>cb</i>	[Fonction]
Specify a procedure that will be called every time LilyPond modifies a grob property. The callback will receive as arguments the grob that is being modified, the name of the C++ file in which the modification was requested, the line number in the C++ file in which the modification was requested, the name of the function in which the modification was requested, the property to be changed, and the new value for the property.	

- ly:set-middle-C!** *context* [Fonction]
Set the `middleCPosition` variable in *context* based on the variables `middleCClefPosition` and `middleCOffset`.
- ly:set-option** *var val* [Fonction]
Set a program option.
- ly:set-property-cache-callback** *cb* [Fonction]
Specify a procedure that will be called whenever lilypond calculates a callback function and caches the result. The callback will receive as arguments the grob whose property it is, the name of the property, the name of the callback that calculated the property, and the new (cached) value of the property.
- ly:simple-closure?** *clos* [Fonction]
Is *clos* a simple closure?
- ly:skyline?** *x* [Fonction]
Is *x* a `Skyline` object?
- ly:skyline-pair?** *x* [Fonction]
Is *x* a `Skyline_pair` object?
- ly:slur-score-count** [Fonction]
count number of slur scores.
- ly:smob-protects** [Fonction]
Return LilyPond's internal smob protection list.
- ly:solve-spring-rod-problem** *springs rods length ragged* [Fonction]
Solve a spring and rod problem for *count* objects, that are connected by *count*-1 *springs*, and an arbitrary number of *rods*. *count* is implicitly given by *springs* and *rods*. The *springs* argument has the format (ideal, inverse_hook) and *rods* is of the form (idx1, idx2, distance).
length is a number, *ragged* a boolean.
The function returns a list containing the force (positive for stretching, negative for compressing and #f for non-satisfied constraints) followed by *spring-count*+1 positions of the objects.
- ly:source-file?** *x* [Fonction]
Is *x* a `Source_file` object?
- ly:spanner?** *g* [Fonction]
Is *g* a spanner object?
- ly:spanner-bound** *spanner dir* [Fonction]
Get one of the bounds of *spanner*. *dir* is -1 for left, and 1 for right.
- ly:spanner-broken-into** *spanner* [Fonction]
Return broken-into list for *spanner*.
- ly:spanner-set-bound!** *spanner dir item* [Fonction]
Set grob *item* as bound in direction *dir* for *spanner*.
- ly:spawn** *command rest* [Fonction]
Simple interface to `g_spawn_sync` *str*. The error is formatted with `format` and *rest*.






ly:spring? <i>x</i>	[Fonction]
Is <i>x</i> a Spring object?	
ly:spring-set-inverse-compress-strength! <i>spring strength</i>	[Fonction]
Set the inverse compress <i>strength</i> of <i>spring</i> .	
ly:spring-set-inverse-stretch-strength! <i>spring strength</i>	[Fonction]
Set the inverse stretch <i>strength</i> of <i>spring</i> .	
ly:staff-symbol-line-thickness <i>grob</i>	[Fonction]
Returns the line-thickness of the staff associated with <i>grob</i> .	
ly:staff-symbol-staff-radius <i>grob</i>	[Fonction]
Returns the radius of the staff associated with <i>grob</i> .	
ly:staff-symbol-staff-space <i>grob</i>	[Fonction]
Returns the staff-space of the staff associated with <i>grob</i> .	
ly:start-environment	[Fonction]
Return the environment (a list of strings) that was in effect at program start.	
ly:stderr-redirect <i>file-name mode</i>	[Fonction]
Redirect stderr to <i>file-name</i> , opened with <i>mode</i> .	
ly:stencil? <i>x</i>	[Fonction]
Is <i>x</i> a Stencil object?	
ly:stencil-add <i>args</i>	[Fonction]
Combine stencils. Takes any number of arguments.	
ly:stencil-aligned-to <i>stil axis dir</i>	[Fonction]
Align <i>stil</i> using its own extents. <i>dir</i> is a number. -1 and 1 are left and right, respectively. Other values are interpolated (so 0 means the center).	
ly:stencil-combine-at-edge <i>first axis direction second padding</i>	[Fonction]
Construct a stencil by putting <i>second</i> next to <i>first</i> . <i>axis</i> can be 0 (x-axis) or 1 (y-axis). <i>direction</i> can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with <i>padding</i> as extra space. <i>first</i> and <i>second</i> may also be '()' or #f.	
ly:stencil-empty? <i>stil</i>	[Fonction]
Return whether <i>stil</i> is empty.	
ly:stencil-expr <i>stil</i>	[Fonction]
Return the expression of <i>stil</i> .	
ly:stencil-extent <i>stil axis</i>	[Fonction]
Return a pair of numbers signifying the extent of <i>stil</i> in <i>axis</i> direction (0 or 1 for x and y axis, respectively).	
ly:stencil-fonts <i>s</i>	[Fonction]
Analyze <i>s</i> , and return a list of fonts used in <i>s</i> .	
ly:stencil-in-color <i>stc r g b</i>	[Fonction]
Put <i>stc</i> in a different color.	
ly:stencil-rotate <i>stil angle x y</i>	[Fonction]
Return a stencil <i>stil</i> rotated <i>angle</i> degrees around the relative offset (x, y). E.g., an offset of (-1, 1) will rotate the stencil around the left upper corner.	

- ly:stencil-rotate-absolute** *stil angle x y* [Fonction]
Return a stencil *stil* rotated *angle* degrees around point (x, y), given in absolute coordinates.
- ly:stencil-scale** *stil x y* [Fonction]
Scale *stil* using the horizontal and vertical scaling factors *x* and *y*.
- ly:stencil-translate** *stil offset* [Fonction]
Return a *stil*, but translated by *offset* (a pair of numbers).
- ly:stencil-translate-axis** *stil amount axis* [Fonction]
Return a copy of *stil* but translated by *amount* in *axis* direction.
- ly:stream-event?** *obj* [Fonction]
Is *obj* a `Stream_event` object?
- ly:string-percent-encode** *str* [Fonction]
Encode all characters in string *str* with hexadecimal percent escape sequences, with the following exceptions: characters -, ., /, and _; and characters in ranges 0-9, A-Z, and a-z.
- ly:string-substitute** *a b s* [Fonction]
Replace string *a* by string *b* in string *s*.
- ly:system-font-load** *name* [Fonction]
Load the OpenType system font '*name.otf*'. Fonts loaded with this command must contain three additional SFNT font tables called LILC, LILF, and LILY, needed for typesetting musical elements. Currently, only the Emmentaler and the Emmentaler-Brace fonts fulfill these requirements.

Note that only `ly:font-get-glyph` and derived code (like `\lookup`) can access glyphs from the system fonts; text strings are handled exclusively via the Pango interface.
- ly:text-interface::interpret-markup** [Fonction]
Convert a text markup into a stencil. Takes three arguments, *layout*, *props*, and *markup*.
layout is a `\layout` block; it may be obtained from a grob with `ly:grob-layout`. *props* is an alist chain, i.e. a list of alists. This is typically obtained with `(ly:grob-alist-chain grob (ly:output-def-lookup layout 'text-font-defaults))`. *markup* is the markup text to be processed.
- ly:translate-cpp-warning-scheme** *str* [Fonction]
Translates a string in C++ printf format and modifies it to use it for scheme formatting.
- ly:translator?** *x* [Fonction]
Is *x* a `Translator` object?
- ly:translator-context** *trans* [Fonction]
Return the context of the translator object *trans*.
- ly:translator-description** *me* [Fonction]
Return an alist of properties of translator *me*.
- ly:translator-group?** *x* [Fonction]
Is *x* a `Translator_group` object?
- ly:translator-name** *trans* [Fonction]
Return the type name of the translator object *trans*. The name is a symbol.
- ly:transpose-key-alist** *l pit* [Fonction]
Make a new key alist of *l* transposed by pitch *pit*.

- ly:truncate-list!** *lst i* [Fonction]
Take at most the first *i* of list *lst*.
- ly:ttf->pfa** *ttf-file-name idx* [Fonction]
Convert the contents of a TrueType font file to PostScript Type 42 font, returning it as a string. The optional *idx* argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of *idx* is 0.
- ly:ttf-ps-name** *ttf-file-name idx* [Fonction]
Extract the PostScript name from a TrueType font. The optional *idx* argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of *idx* is 0.
- ly:undead?** *x* [Fonction]
Is *x* a **Undead** object?
- ly:unit** [Fonction]
Return the unit used for lengths as a string.
- ly:unpure-pure-container?** *clos* [Fonction]
Is *clos* an unpure pure container?
- ly:unpure-pure-container-pure-part** *pc* [Fonction]
Return the pure part of *pc*.
- ly:unpure-pure-container-unpure-part** *pc* [Fonction]
Return the unpure part of *pc*.
- ly:usage** [Fonction]
Print usage message.
- ly:verbose-output?** [Fonction]
Was verbose output requested, i.e. loglevel at least **DEBUG**?
- ly:version** [Fonction]
Return the current lilypond version as a list, e.g., (1 3 127 uu1).
- ly:warning** *str rest* [Fonction]
A Scheme callable function to issue the warning *str*. The message is formatted with **format** and *rest*.
- ly:warning-located** *location str rest* [Fonction]
A Scheme callable function to issue the warning *str* at the specified location in an input file. The message is formatted with **format** and *rest*.
- ly:wide-char->utf-8** *wc* [Fonction]
Encode the Unicode codepoint *wc*, an integer, as UTF-8.

Annexe B Aide-mémoire

Syntaxe	Description	Exemple
<code>1 2 8 16</code>	valeurs rythmiques	
<code>c4. c4..</code>	notes pointées	
<code>c d e f g a b</code>	gamme	
<code>fis bes</code>	altérations	
<code>\clef treble \clef bass</code>	clés	
<code>\time 3/4 \time 4/4</code>	chiffre de mesure, métrique	
<code>r4 r8</code>	silences	
<code>d ~ d</code>	liaison de tenue	
<code>\key es \major</code>	armure	

`note'`

monter d'une octave

`note,`

baisser d'une octave

`c(d e)`

liaisons

`c\ (c(d) e\)`

liaisons de phrasé

`a8[b]`

ligatures

`<< \new Staff ... >>`

ajouter des portées

`c-> c-.`

indications d'articulation

`c2\mf c\s fz`

nuances

`a\< a a\!`

crescendo



`a\> a a\!`

decrescendo

`< >`

accords

`\partial 8`

levées, anacrouses

`\times 2/3 {f g a}`

triolet

`\grace`

appogiatures

`\lyricmode { twinkle }`

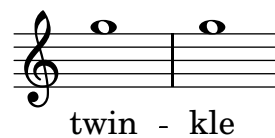
ajouter des paroles

twinkle

`\new Lyrics`

imprimer les paroles

twinkle

`twin -- kle`diviser un mot en
plusieurs syllabes`\chordmode { c:dim f:maj7 }`

accords chiffrés

`\context ChordNames`imprimer les chiffrages
d'accordsC^o F^Δ`<<{e f} \ \ {c d}>>`

polyphonie

`s4 s8 s16`

silences invisibles

Annexe C Grammaire de LilyPond

Cette annexe présente la grammaire du langage de LilyPond, telle qu'elle ressort de l'analyseur syntaxique.

Grammar

```

1 start_symbol: lilypond

3 start_symbol: "#{ " embedded_lilypond

4 lilypond: /* empty */
5           | lilypond toplevel_expression
6           | lilypond assignment
7           | lilypond error
8           | lilypond "\version-error"

10 toplevel_expression: lilypond_header
11                     | book_block
12                     | bookpart_block
13                     | score_block
14                     | composite_music
15                     | full_markup
16                     | full_markup_list
17                     | output_def

18 embedded_scm_bare: SCM_TOKEN
19                  | SCM_IDENTIFIER

20 embedded_scm_bare_arg: embedded_scm_bare
21                      | STRING
22                      | STRING_IDENTIFIER
23                      | full_markup
24                      | full_markup_list
25                      | context_modification
26                      | score_block
27                      | context_def_spec_block
28                      | book_block
29                      | bookpart_block
30                      | output_def

31 embedded_scm: embedded_scm_bare
32              | scm_function_call

33 embedded_scm_arg: embedded_scm_bare_arg
34                  | scm_function_call
35                  | music_arg

36 scm_function_call: SCM_FUNCTION function_arglist

37 embedded_lilypond: /* empty */
38                  | identifier_init
39                  | music_embedded music_embedded music_list

```



```

40          | error
41          | "\version-error" embedded_lilypond

42 lilypond_header_body: /* empty */
43          | lilypond_header_body assignment

44 lilypond_header: "\header" '{' lilypond_header_body '}'

45 assignment_id: STRING
46          | LYRICS_STRING

47 assignment: assignment_id '=' identifier_init
48          | assignment_id property_path '=' identifier_init
49          | embedded_scm

50 identifier_init: score_block
51          | book_block
52          | bookpart_block
53          | output_def
54          | context_def_spec_block
55          | music_assign
56          | post_event_nofinger
57          | number_expression
58          | FRACTION
59          | string
60          | embedded_scm
61          | full_markup
62          | full_markup_list
63          | context_modification

64 context_def_spec_block: "\context" '{' context_def_spec_body '}'

65 context_mod_arg: embedded_scm
66          | composite_music

67 context_mod_embedded: context_mod_arg

68 context_def_spec_body: /* empty */
69          | CONTEXT_DEF_IDENTIFIER
70          | context_def_spec_body context_mod
71          | context_def_spec_body context_modification
72          | context_def_spec_body context_mod_embedded

73 book_block: "\book" '{' book_body '}'

74 book_body: /* empty */
75          | BOOK_IDENTIFIER
76          | book_body paper_block
77          | book_body bookpart_block
78          | book_body score_block
79          | book_body composite_music
80          | book_body full_markup

```

```

81          | book_body full_markup_list

83 book_body: book_body lilypond_header
84          | book_body embedded_scm
85          | book_body error

86 bookpart_block: "\bookpart" '{' bookpart_body '}'

87 bookpart_body: /* empty */
88              | BOOK_IDENTIFIER
89              | bookpart_body paper_block
90              | bookpart_body score_block
91              | bookpart_body composite_music
92              | bookpart_body full_markup
93              | bookpart_body full_markup_list

95 bookpart_body: bookpart_body lilypond_header
96              | bookpart_body embedded_scm
97              | bookpart_body error

98 score_block: "\score" '{' score_body '}'

99 score_body: music
100          | SCORE_IDENTIFIER

102 score_body: score_body lilypond_header
103          | score_body output_def
104          | score_body error

105 paper_block: output_def

106 output_def: output_def_body '}'

107 output_def_head: "\paper"
108                | "\midi"
109                | "\layout"

110 output_def_head_with_mode_switch: output_def_head

111 music_or_context_def: music_arg

113 music_or_context_def: "\context" '{' context_def_spec_body '}'

114 output_def_body: output_def_head_with_mode_switch '{'
115                | output_def_head_with_mode_switch
116                  '{'
117                    OUTPUT_DEF_IDENTIFIER
118                  | output_def_body assignment

118 output_def_body: output_def_body music_or_context_def
119                | output_def_body error

```

```

120 tempo_event: "\tempo" steno_duration '=' tempo_range
121             | "\tempo" scalar_closed steno_duration '=' tempo_range
122             | "\tempo" scalar

123 music_list: /* empty */
124             | music_list music_embedded
125             | music_list error

126 braced_music_list: '{' music_list '}'

127 music: music_arg
128       | lyric_element_music

129 music_embedded: music
130               | embedded_scm

131 music_arg: simple_music
132           | composite_music

133 music_assign: simple_music
134             | composite_music
135             | lyric_element_music

136 repeated_music: "\repeat" simple_string unsigned_number music
137               | "\repeat"
                  simple_string
                  unsigned_number
                  music
                  "\alternative"
                  braced_music_list

138 sequential_music: "\sequential" braced_music_list
139                 | braced_music_list

140 simultaneous_music: "\simultaneous" braced_music_list
141                   | "<<" music_list ">>"

142 simple_music: event_chord
143             | music_property_def
144             | context_change

146 context_modification: "\with" '{' context_mod_list '}'
147                   | "\with" CONTEXT_MOD_IDENTIFIER
148                   | CONTEXT_MOD_IDENTIFIER
149                   | "\with" embedded_scm_closed

150 optional_context_mod: /* empty */
151                     | context_modification

152 context_mod_list: /* empty */
153                 | context_mod_list context_mod
154                 | context_mod_list CONTEXT_MOD_IDENTIFIER

```



```

177 function_arglist_closed_nonbackup: function_arglist_nonbackup_common
178                                     | "optional?"
                                         "scheme?"
                                         function_arglist
                                         embedded_scm_arg_closed
179                                     | "optional?"
                                         "scheme?"
                                         function_arglist_closed
                                         bare_number_closed

180 function_arglist_nonbackup: function_arglist_nonbackup_common
181                                     | "optional?"
                                         "scheme?"
                                         function_arglist
                                         embedded_scm_arg
182                                     | "optional?"
                                         "scheme?"
                                         function_arglist_closed
                                         bare_number

183 function_arglist_keep: function_arglist_common
184                             | function_arglist_backup

185 function_arglist_closed_keep: function_arglist_closed_common
186                             | function_arglist_backup

187 function_arglist_backup: "optional?"
                             "scheme?"
                             function_arglist_keep
                             embedded_scm_arg_closed
188                             | "optional?"
                             "scheme?"
                             function_arglist_closed_keep
                             post_event_nofinger
189                             | "optional?"
                             "scheme?"
                             function_arglist_keep
                             lyric_element
190                             | "optional?"
                             "scheme?"
                             function_arglist_closed_keep
                             UNSIGNED
191                             | "optional?"
                             "scheme?"
                             function_arglist_closed_keep
                             REAL
192                             | "optional?"
                             "scheme?"
                             function_arglist_closed_keep
                             NUMBER_IDENTIFIER
193                             | "optional?"

```

```

        "scheme?"
        function_arglist_closed_keep
        FRACTION
194 | "optional?"
        "scheme?"
        function_arglist_closed_keep
        '-'
        UNSIGNED
195 | "optional?"
        "scheme?"
        function_arglist_closed_keep
        '-'
        REAL
196 | "optional?"
        "scheme?"
        function_arglist_closed_keep
        '-'
        NUMBER_IDENTIFIER
197 | "optional?"
        "ly:pitch?"
        function_arglist_keep
        pitch_also_in_chords
198 | "optional?"
        "ly:duration?"
        function_arglist_closed_keep
        duration_length
199 | "optional?"
        "scheme?"
        function_arglist_backup
        "(backed-up?)"
200 | function_arglist_backup
        "(reparsed?)"
        embedded_scm_arg_closed
201 | function_arglist_backup
        "(reparsed?)"
        bare_number

202 function_arglist: function_arglist_common
203 | function_arglist_nonbackup

204 function_arglist_common: function_arglist_bare
205 | "scheme?"
        function_arglist_optional
        embedded_scm_arg
206 | "scheme?"
        function_arglist_closed_optional
        bare_number
207 | "scheme?"
        function_arglist_closed_optional
        FRACTION
208 | "scheme?"
        function_arglist_closed_optional

```

```

                                post_event_nofinger
209                             | function_arglist_common_minus
210                             | function_arglist_common_lyric

211 function_arglist_common_lyric: "scheme?"
                                function_arglist_optional
                                lyric_element
212                             | function_arglist_common_lyric
                                "(reparsed?)"
                                lyric_element_arg

213 function_arglist_common_minus: "scheme?"
                                function_arglist_closed_optional
                                '-'
                                UNSIGNED
214                             | "scheme?"
                                function_arglist_closed_optional
                                '-'
                                REAL
215                             | "scheme?"
                                function_arglist_closed_optional
                                '-'
                                NUMBER_IDENTIFIER
216                             | function_arglist_common_minus
                                "(reparsed?)"
                                bare_number

217 function_arglist_closed: function_arglist_closed_common
218                             | function_arglist_closed_nonbackup

219 function_arglist_closed_common: function_arglist_bare
220                             | "scheme?"
                                function_arglist_optional
                                embedded_scm_arg_closed
221                             | "scheme?"
                                function_arglist_closed_optional
                                bare_number
222                             | "scheme?"
                                function_arglist_closed_optional
                                '-'
                                UNSIGNED
223                             | "scheme?"
                                function_arglist_closed_optional
                                '-'
                                REAL
224                             | "scheme?"
                                function_arglist_closed_optional
                                '-'
                                NUMBER_IDENTIFIER
225                             | "scheme?"
                                function_arglist_closed_optional
                                post_event_nofinger

```

```

226             | "scheme?"
                function_arglist_closed_optional
                FRACTION
227             | "scheme?"
                function_arglist_optional
                lyric_element

228 function_arglist_optional: function_arglist_keep
229             | function_arglist_backup "(backed-up?)"
230             | "optional?"
                "ly:pitch?"
                function_arglist_optional
231             | "optional?"
                "ly:duration?"
                function_arglist_optional

232 function_arglist_closed_optional: function_arglist_closed_keep
233             | function_arglist_backup
                "(backed-up?)"
234             | "optional?"
                "ly:pitch?"
                function_arglist_closed_optional
235             | "optional?"
                "ly:duration?"
                function_arglist_closed_optional

236 embedded_scm_closed: embedded_scm_bare
237             | scm_function_call_closed

238 embedded_scm_arg_closed: embedded_scm_bare_arg
239             | scm_function_call_closed
240             | closed_music

241 scm_function_call_closed: SCM_FUNCTION function_arglist_closed

242 function_arglist_bare: EXPECT_NO_MORE_ARGS
243             | "ly:pitch?"
                function_arglist_optional
                pitch_also_in_chords
244             | "ly:duration?"
                function_arglist_closed_optional
                duration_length
245             | "optional?"
                "ly:pitch?"
                function_arglist_skip
                "\default"
246             | "optional?"
                "ly:duration?"
                function_arglist_skip
                "\default"
247             | "optional?"
                "scheme?"

```



```

                                function_arglist_skip
                                "\default"

248 music_function_call: MUSIC_FUNCTION function_arglist

249 optional_id: /* empty */
250             | '=' simple_string

251 complex_music: music_function_call
252             | repeated_music
253             | re_rhythmed_music
254             | complex_music_prefix music

255 complex_music_prefix: "\context"
                        simple_string
                        optional_id
                        optional_context_mod
256             | "\new"
                        simple_string
                        optional_id
                        optional_context_mod

257 mode_changed_music: mode_changing_head grouped_music_list
258             | mode_changing_head_with_context
                        optional_context_mod
                        grouped_music_list

259 mode_changing_head: "\notemode"
260             | "\drummode"
261             | "\figuremode"
262             | "\chordmode"
263             | "\lyricmode"

264 mode_changing_head_with_context: "\drums"
265                               | "\figures"
266                               | "\chords"
267                               | "\lyrics"

269 new_lyrics: "\addlyrics" composite_music

271 new_lyrics: new_lyrics "\addlyrics" 0 composite_music

272 re_rhythmed_music: composite_music new_lyrics

274 re_rhythmed_music: "\lyricsto" simple_string 1 music

275 context_change: "\change" STRING '=' STRING

276 property_path_revved: embedded_scm_closed
277             | property_path_revved embedded_scm_closed

278 property_path: property_path_revved

```

```

279 property_operation: STRING '=' scalar
280             | "\unset" simple_string
281             | "\override" simple_string property_path '=' scalar
282             | "\revert" simple_string embedded_scm

283 context_def_mod: "\consists"
284             | "\remove"
285             | "\accepts"
286             | "\defaultchild"
287             | "\denies"
288             | "\alias"
289             | "\type"
290             | "\description"
291             | "\name"

292 context_mod: property_operation
293             | context_def_mod STRING
294             | context_def_mod embedded_scm

295 context_prop_spec: simple_string
296             | simple_string '.' simple_string

297 simple_music_property_def: "\override"
                           context_prop_spec
                           property_path
                           '='
                           scalar
298             | "\revert" context_prop_spec embedded_scm
299             | "\set" context_prop_spec '=' scalar
300             | "\unset" context_prop_spec

301 music_property_def: simple_music_property_def

302 string: STRING
303             | STRING_IDENTIFIER
304             | string '+' string

305 simple_string: STRING
306             | LYRICS_STRING
307             | STRING_IDENTIFIER

308 scalar: embedded_scm_arg
309             | bare_number
310             | FRACTION
311             | lyric_element

312 scalar_closed: embedded_scm_arg_closed
313             | bare_number
314             | FRACTION
315             | lyric_element

```

```

316 event_chord: simple_element post_events
317             | simple_chord_elements post_events
318             | CHORD_REPETITION optional_notemode_duration post_events
319             | MULTI_MEASURE_REST optional_notemode_duration post_events
320             | command_element
321             | note_chord_element

322 note_chord_element: chord_body optional_notemode_duration post_events

323 chord_body: "<" chord_body_elements ">"

324 chord_body_elements: /* empty */
325                     | chord_body_elements chord_body_element

326 chord_body_element: pitch
                     | exclamations
                     | questions
                     | octave_check
                     | post_events
327                     | DRUM_PITCH post_events
328                     | music_function_chord_body

329 music_function_chord_body: music_function_call
330                           | MUSIC_IDENTIFIER

331 music_function_call_closed: MUSIC_FUNCTION function_arglist_closed

332 event_function_event: EVENT_FUNCTION function_arglist_closed

333 command_element: command_event
334                 | "\"["
335                 | "\"]"
336                 | "\"
337                 | "'"

338 command_event: "\\~"
339               | tempo_event

340 post_events: /* empty */
341             | post_events post_event

342 post_event_nofinger: direction_less_event
343                     | script_dir music_function_call_closed
344                     | "---"
345                     | "--"
346                     | script_dir direction_reqd_event
347                     | script_dir direction_less_event
348                     | '^' fingering
349                     | '_' fingering

350 post_event: post_event_nofinger
351            | '-' fingering

```

```

352 string_number_event: E_UNSIGNED

353 direction_less_char: '['
354                     | ']'
355                     | '~'
356                     | '('
357                     | ')'
358                     | "\!"
359                     | "\"("
360                     | "\"\"
361                     | "\">"
362                     | "\"<"

363 direction_less_event: direction_less_char
364                     | string_number_event
365                     | EVENT_IDENTIFIER
366                     | tremolo_type
367                     | event_function_event

368 direction_reqd_event: gen_text_def
369                     | script_abbreviation

370 octave_check: /* empty */
371             | '='
372             | '=' sub_quotes
373             | '=' sup_quotes

374 sup_quotes: '''
375           | sup_quotes '''

376 sub_quotes: ','
377           | sub_quotes ','

378 steno_pitch: NOTENAME_PITCH
379           | NOTENAME_PITCH sup_quotes
380           | NOTENAME_PITCH sub_quotes

381 steno_tonic_pitch: TONICNAME_PITCH
382           | TONICNAME_PITCH sup_quotes
383           | TONICNAME_PITCH sub_quotes

384 pitch: steno_pitch
385       | PITCH_IDENTIFIER

386 pitch_also_in_chords: pitch
387                   | steno_tonic_pitch

388 gen_text_def: full_markup
389             | simple_string

390 fingering: UNSIGNED

```

```

391 script_abbreviation: '^'
392                      | '+'
393                      | '-'
394                      | '|'
395                      | ">"
396                      | '.'
397                      | '_'

398 script_dir: '_'
399           | '^'
400           | '-'

401 duration_length: multiplied_duration

402 maybe_notemode_duration: /* empty */
403                        | multiplied_duration

404 optional_notemode_duration: maybe_notemode_duration

405 steno_duration: bare_unsigned dots
406                | DURATION_IDENTIFIER dots

407 multiplied_duration: steno_duration
408                    | multiplied_duration '*' bare_unsigned
409                    | multiplied_duration '*' FRACTION

410 dots: /* empty */
411      | dots '.'

412 tremolo_type: ':'
413             | ':' bare_unsigned

414 bass_number: UNSIGNED
415            | STRING
416            | full_markup

417 figured_bass_alteration: '-'
418                        | '+'
419                        | '!'

420 bass_figure: "_"
421            | bass_number
422            | bass_figure ']'
423            | bass_figure figured_bass_alteration
424            | bass_figure figured_bass_modification

425 figured_bass_modification: "\+"
426                          | "\!"
427                          | '/'
428                          | "\"

```

```

429 br_bass_figure: bass_figure
430           | '[' bass_figure

431 figure_list: /* empty */
432           | figure_list br_bass_figure

433 figure_spec: FIGURE_OPEN figure_list FIGURE_CLOSE

434 optional_rest: /* empty */
435           | "\rest"

436 simple_element: pitch
                  exclamations
                  questions
                  octave_check
                  maybe_notemode_duration
                  optional_rest
437           | DRUM_PITCH optional_notemode_duration
438           | RESTNAME optional_notemode_duration

439 simple_chord_elements: new_chord
440           | figure_spec optional_notemode_duration

441 lyric_element: lyric_markup
442           | LYRICS_STRING

443 lyric_element_arg: lyric_element
444           | lyric_element multiplied_duration post_events
445           | lyric_element post_event post_events
446           | LYRIC_ELEMENT optional_notemode_duration post_events

447 lyric_element_music: lyric_element
                  optional_notemode_duration
                  post_events

448 new_chord: steno_tonic_pitch optional_notemode_duration
449           | steno_tonic_pitch
                  optional_notemode_duration
                  chord_separator
                  chord_items

450 chord_items: /* empty */
451           | chord_items chord_item

452 chord_separator: ":"
453           | "^"
454           | "/" steno_tonic_pitch
455           | "/" steno_tonic_pitch

456 chord_item: chord_separator
457           | step_numbers
458           | CHORD_MODIFIER

```

```
459 step_numbers: step_number
460             | step_numbers '.' step_number

461 step_number: bare_unsigned
462           | bare_unsigned '+'
463           | bare_unsigned "-"

464 tempo_range: bare_unsigned
465           | bare_unsigned '~' bare_unsigned

466 number_expression: number_expression '+' number_term
467                 | number_expression '-' number_term
468                 | number_term

469 number_term: number_factor
470           | number_factor '*' number_factor
471           | number_factor '/' number_factor

472 number_factor: '-' number_factor
473             | bare_number

474 bare_number: bare_number_closed
475           | UNSIGNED NUMBER_IDENTIFIER
476           | REAL NUMBER_IDENTIFIER

477 bare_number_closed: UNSIGNED
478                 | REAL
479                 | NUMBER_IDENTIFIER

480 bare_unsigned: UNSIGNED

481 unsigned_number: UNSIGNED
482               | NUMBER_IDENTIFIER

483 exclamations: /* empty */
484             | exclamations '!'

485 questions: /* empty */
486           | questions '?'

487 lyric_markup: LYRIC_MARKUP_IDENTIFIER

489 lyric_markup: LYRIC_MARKUP 2 markup_top

490 full_markup_list: MARKUPLIST_IDENTIFIER

492 full_markup_list: "\markuplist" 3 markup_list

493 full_markup: MARKUP_IDENTIFIER

495 full_markup: "\markup" 4 markup_top
```

```

496 markup_top: markup_list
497           | markup_head_1_list simple_markup
498           | simple_markup

500 markup_scm: embedded_scm_bare 5 "(backed-up?)"

501 markup_list: MARKUPLIST_IDENTIFIER
502           | markup_composed_list
503           | markup_braced_list
504           | markup_command_list
505           | markup_scm MARKUPLIST_IDENTIFIER

506 markup_composed_list: markup_head_1_list markup_braced_list

507 markup_braced_list: '{' markup_braced_list_body '}'

508 markup_braced_list_body: /* empty */
509                       | markup_braced_list_body markup
510                       | markup_braced_list_body markup_list

511 markup_command_list: MARKUP_LIST_FUNCTION markup_command_list_arguments

512 markup_command_basic_arguments: "markup-list?"
                                markup_command_list_arguments
                                markup_list
513                       | "scheme?"
                                markup_command_list_arguments
                                embedded_scm_closed
514                       | EXPECT_NO_MORE_ARGS

515 markup_command_list_arguments: markup_command_basic_arguments
516                       | "markup?"
                                markup_command_list_arguments
                                markup

517 markup_head_1_item: MARKUP_FUNCTION
                    "markup?"
                    markup_command_list_arguments

518 markup_head_1_list: markup_head_1_item
519           | markup_head_1_list markup_head_1_item

520 simple_markup: STRING
521           | MARKUP_IDENTIFIER
522           | LYRIC_MARKUP_IDENTIFIER
523           | STRING_IDENTIFIER

525 simple_markup: "\\score" 6 '{' score_body '}'
526           | MARKUP_FUNCTION markup_command_basic_arguments
527           | markup_scm MARKUP_IDENTIFIER

```



```

528 markup: markup_head_1_list simple_markup
529         | simple_markup

```

Terminals, with rules where they appear

```

182 187 188 189 190 191 192 193 194 195 196 197 198 199 230 231
190 191 192 193 194 195 196 199 205 206 207 208 211 213 214 215
220 221 222 223 224 225 226 227 247 513
234 235 245 246 247
417 467 472
"#{" (352) 3
"(backed-up?)" (347) 199 229 233 500
"(reparsed?)" (348) 200 201 212 216
"--" (317) 463
"---" (338) 344
"/" (318) 454
"/+" (314) 455
":" (316) 452
"<" (319) 323
"<<" (321) 141
">" (320) 323 395
">>" (322) 141
"!\" (327) 358 426
\" (323) 336 428
\"(\" (329) 359
\"\\\" (326) 360
\"\\+\" (332) 425
\"\\<\" (331) 362
\"\\>\" (324) 361
\"\\[\" (328) 334
\"\\]\" (330) 335
\"\\accepts\" (273) 285
\"\\addlyrics\" (262) 269 271
\"\\alias\" (274) 288
\"\\alternative\" (260) 137
\"\\book\" (275) 73
\"\\bookpart\" (276) 86
\"\\C[haracter]\" (325)
\"\\change\" (277) 275
\"\\chordmode\" (278) 262
\"\\chords\" (279) 266
\"\\consists\" (280) 283
\"\\context\" (281) 64 113 255
\"\\default\" (263) 245 246 247
\"\\defaultchild\" (282) 286
\"\\denies\" (283) 287
\"\\description\" (284) 290
\"\\drummode\" (285) 260
\"\\drums\" (286) 264

```

```

"\figuremode" (287) 261
"\figures" (288) 265
"\header" (289) 44
"\layout" (291) 109
"\lyricmode" (292) 263
"\lyrics" (293) 267
"\lyricsto" (294) 274
"\markup" (295) 495
"\markuplist" (296) 492
"\midi" (297) 108
"\name" (298) 291
"\new" (313) 256
"\notemode" (299) 259
"\override" (300) 281 297
"\paper" (301) 107
"\remove" (302) 284
"\repeat" (259) 136 137
"\rest" (303) 435
"\revert" (304) 282 298
"\score" (305) 98 525
"\sequential" (306) 138
"\set" (307) 299
"\simultaneous" (308) 140
"\tempo" (309) 120 121 122
"\type" (310) 289
"\unset" (311) 280 300
"\version-error" (290) 8 41
"\with" (312) 146 147 149
"\~" (333) 338
"^" (315) 453
"_" (337) 420
"__" (334) 345
"end of input" (0) 0
"ly:duration?" (345) 168 171 198 231 235 244 246
"ly:pitch?" (344) 167 170 197 230 234 243 245
"markup-list?" (349) 512
"markup?" (343) 516 517
"optional?" (350) 167 168 169 170 171 172 173 174 175 176 178 179 181
"scheme?" (346) 169 172 173 174 175 176 178 179 181 182 187 188 189
'!' (33) 419 484
''' (39) 374 375
'(' (40) 356
')' (41) 357
'*' (42) 408 409 470
'+' (43) 304 392 418 462 466
',' (44) 376 377
'-' (45) 174 175 176 194 195 196 213 214 215 222 223 224 351 393 400
'.' (46) 296 396 411 460
'/' (47) 427 471
':' (58) 412 413
'=' (61) 47 48 120 121 250 275 279 281 297 299 371 372 373
'?' (63) 486

```

'[' (91) 353 430
 ']' (93) 354 422
 '^' (94) 348 391 399
 '_' (95) 349 397 398
 '{' (123) 44 64 73 86 98 113 114 115 126 146 507 525
 '|' (124) 337 394
 '}' (125) 44 64 73 86 98 106 113 126 146 507 525
 '~' (126) 355 465
 BOOK_IDENTIFIER (353) 75 88
 CHORD_MODIFIER (355) 458
 CHORD_REPETITION (356) 318
 CHORDMODIFIER_PITCH (354)
 CHORDMODIFIERS (339)
 COMPOSITE (261)
 CONTEXT_DEF_IDENTIFIER (357) 69
 CONTEXT_MOD_IDENTIFIER (358) 147 148 154
 DRUM_PITCH (359) 327 437
 DURATION_IDENTIFIER (265) 406
 E_UNSIGNED (342) 352
 error (256) 7 40 85 97 104 119 125
 EVENT_FUNCTION (361) 332
 EVENT_IDENTIFIER (360) 365
 EXPECT_NO_MORE_ARGS (351) 242 514
 FIGURE_CLOSE (335) 433
 FIGURE_OPEN (336) 433
 FRACTION (362) 58 172 193 207 226 310 314 409
 FUNCTION_ARGLIST (264)
 LYRIC_ELEMENT (364) 446
 LYRIC_MARKUP (340) 489
 LYRIC_MARKUP_IDENTIFIER (365) 487 522
 LYRICS_STRING (363) 46 306 442
 MARKUP_FUNCTION (366) 517 526
 MARKUP_IDENTIFIER (368) 493 521 527
 MARKUP_LIST_FUNCTION (367) 511
 MARKUPLIST_IDENTIFIER (369) 490 501 505
 MULTI_MEASURE_REST (341) 319
 MUSIC_FUNCTION (370) 248 331
 MUSIC_IDENTIFIER (371) 162 330
 NOTENAME_PITCH (269) 378 379 380
 NUMBER_IDENTIFIER (271) 176 192 196 215 224 475 476 479 482
 OUTPUT_DEF_IDENTIFIER (372) 115
 PITCH_IDENTIFIER (270) 385
 PREC_BOT (258)
 PREC_TOP (272)
 REAL (266) 175 191 195 214 223 476 478
 RESTNAME (373) 438
 SCM_FUNCTION (374) 36 241
 SCM_IDENTIFIER (375) 19
 SCM_TOKEN (376) 18
 SCORE_IDENTIFIER (377) 100
 STRING (378) 21 45 275 279 293 302 305 415 520
 STRING_IDENTIFIER (379) 22 303 307 523

TONICNAME_PITCH (268) 381 382 383
 UNARY_MINUS (380)
 UNSIGNED (267) 174 190 194 213 222 390 414 475 477 480 481

Nonterminals, with rules where they appear

assignment (162)
 on left: 47 48 49, on right: 6 43 116
assignment_id (161)
 on left: 45 46, on right: 47 48
bare_number_closed (307)
 on left: 477 478 479, on right: 179 474
bass_number (283)
 on left: 414 415 416, on right: 421
book_block (168)
 on left: 73, on right: 11 28 51
bookpart_block (171)
 on left: 86, on right: 12 29 52 77
br_bass_figure (287)
 on left: 429 430, on right: 432
braced_music_list (187)
 on left: 126, on right: 137 138 139 140
chord_body (250)
 on left: 323, on right: 322
chord_body_element (252)
 on left: 326 327 328, on right: 325
chord_body_elements (251)
 on left: 324 325, on right: 323 325
chord_item (299)
 on left: 456 457 458, on right: 451
chord_items (297)
 on left: 450 451, on right: 449 451
chord_separator (298)
 on left: 452 453 454 455, on right: 449 456
closed_music (201)
 on left: 158 159 160, on right: 159 240
command_element (256)
 on left: 333 334 335 336 337, on right: 320
command_event (257)
 on left: 338 339, on right: 333
complex_music (225)
 on left: 251 252 253 254, on right: 156
complex_music_prefix (226)
 on left: 255 256, on right: 159 254
context_change (235)
 on left: 275, on right: 144
context_def_spec_block (164)
 on left: 64, on right: 27 54
context_mod (240)
 on left: 292 293 294, on right: 70 153
context_mod_arg (165)

- on left: 65 66, on right: 67
- context_mod_embedded (166)
 - on left: 67, on right: 72 155
- context_prop_spec (241)
 - on left: 295 296, on right: 297 298 299 300
- direction_reqd_event (264)
 - on left: 368 369, on right: 346
- dots (281)
 - on left: 410 411, on right: 405 406 411
- duration_length (276)
 - on left: 401, on right: 171 198 244
- embedded_lilypond (158)
 - on left: 37 38 39 40 41, on right: 3 41
- embedded_scm_arg (156)
 - on left: 33 34 35, on right: 181 205 308
- embedded_scm_bare (153)
 - on left: 18 19, on right: 20 31 236 500
- embedded_scm_closed (219)
 - on left: 236 237, on right: 149 276 277 513
- event_chord (248)
 - on left: 316 317 318 319 320 321, on right: 142
- event_function_event (255)
 - on left: 332, on right: 367
- exclamations (310)
 - on left: 483 484, on right: 326 436 484
- figure_list (288)
 - on left: 431 432, on right: 432 433
- figure_spec (289)
 - on left: 433, on right: 440
- figured_bass_alteration (284)
 - on left: 417 418 419, on right: 423
- figured_bass_modification (286)
 - on left: 425 426 427 428, on right: 424
- fingering (273)
 - on left: 390, on right: 348 349 351
- full_markup (316)
 - on left: 493 495, on right: 15 23 61 80 92 388 416
- full_markup_list (314)
 - on left: 490 492, on right: 16 24 62 81 93
- function_arglist (211)
 - on left: 202 203, on right: 36 170 178 181 248
- function_arglist_common_lyric (213)
 - on left: 211 212, on right: 210 212
- function_arglist_keep (208)
 - on left: 183 184, on right: 187 189 197 228
- function_arglist_nonbackup (207)
 - on left: 180 181 182, on right: 203
- gen_text_def (272)
 - on left: 388 389, on right: 368
- grouped_music_list (203)
 - on left: 164 165, on right: 163 257 258
- lilypond (150)

```

    on left: 4 5 6 7 8, on right: 1 5 6 7 8
lilypond_header (160)
    on left: 44, on right: 10 83 95 102
lilypond_header_body (159)
    on left: 42 43, on right: 43 44
lyric_element_arg (294)
    on left: 443 444 445 446, on right: 212
lyric_element_music (295)
    on left: 447, on right: 128 135
lyric_markup (312)
    on left: 487 489, on right: 441
markup (332)
    on left: 528 529, on right: 509 516
markup_braced_list (323)
    on left: 507, on right: 503 506
markup_braced_list_body (324)
    on left: 508 509 510, on right: 507 509 510
markup_command_list (325)
    on left: 511, on right: 504
markup_composed_list (322)
    on left: 506, on right: 502
markup_head_1_item (328)
    on left: 517, on right: 518 519
markup_head_1_list (329)
    on left: 518 519, on right: 497 506 519 528
markup_list (321)
    on left: 501 502 503 504 505, on right: 492 496 510 512
markup_scm (319)
    on left: 500, on right: 505 527
markup_top (318)
    on left: 496 497 498, on right: 489 495
maybe_notemode_duration (277)
    on left: 402 403, on right: 404 436
mode_changed_music (227)
    on left: 257 258, on right: 161
mode_changing_head (228)
    on left: 259 260 261 262 263, on right: 257
music (188)
    on left: 127 128, on right: 99 129 136 137 254 274
music_arg (190)
    on left: 131 132, on right: 35 111 127
music_assign (191)
    on left: 133 134 135, on right: 55
music_bare (202)
    on left: 161 162 163, on right: 157 158
music_embedded (189)
    on left: 129 130, on right: 39 124
music_function_call (223)
    on left: 248, on right: 251 329
music_function_call_closed (254)
    on left: 331, on right: 160 343
music_function_chord_body (253)

```

- on left: 329 330, on right: 328
- music_list (186)
 - on left: 123 124 125, on right: 39 124 125 126 141
- music_or_context_def (181)
 - on left: 111 113, on right: 118
- music_property_def (243)
 - on left: 301, on right: 143
- new_chord (296)
 - on left: 448 449, on right: 439
- new_lyrics (230)
 - on left: 269 271, on right: 271 272
- note_chord_element (249)
 - on left: 322, on right: 321
- number_expression (303)
 - on left: 466 467 468, on right: 57 466 467
- number_factor (305)
 - on left: 472 473, on right: 469 470 471 472
- number_term (304)
 - on left: 469 470 471, on right: 466 467 468
- octave_check (265)
 - on left: 370 371 372 373, on right: 326 436
- optional_context_mod (198)
 - on left: 150 151, on right: 255 256 258
- optional_id (224)
 - on left: 249 250, on right: 255 256
- optional_rest (290)
 - on left: 434 435, on right: 436
- output_def (178)
 - on left: 106, on right: 17 30 53 103 105
- output_def_head (179)
 - on left: 107 108 109, on right: 110
- output_def_head_with_mode_switch (180)
 - on left: 110, on right: 114 115
- paper_block (177)
 - on left: 105, on right: 76 89
- pitch (270)
 - on left: 384 385, on right: 326 386 436
- pitch_also_in_chords (271)
 - on left: 386 387, on right: 170 197 243
- post_event (260)
 - on left: 350 351, on right: 341 445
- property_operation (238)
 - on left: 279 280 281 282, on right: 292
- property_path (237)
 - on left: 278, on right: 48 281 297
- property_path_revved (236)
 - on left: 276 277, on right: 277 278
- questions (311)
 - on left: 485 486, on right: 326 436 486
- re_rhythmed_music (233)
 - on left: 272 274, on right: 253
- repeated_music (192)

- on left: 136 137, on right: 252
- scalar (246)
 - on left: 308 309 310 311, on right: 122 279 281 297 299
- scalar_closed (247)
 - on left: 312 313 314 315, on right: 121
- scm_function_call (157)
 - on left: 36, on right: 32 34
- scm_function_call_closed (221)
 - on left: 241, on right: 237 239
- score_block (174)
 - on left: 98, on right: 13 26 50 78 90
- script_dir (275)
 - on left: 398 399 400, on right: 343 346 347
- sequential_music (193)
 - on left: 138 139, on right: 165
- simple_chord_elements (292)
 - on left: 439 440, on right: 317
- simple_element (291)
 - on left: 436 437 438, on right: 316
- simple_music (195)
 - on left: 142 143 144, on right: 131 133
- simple_music_property_def (242)
 - on left: 297 298 299 300, on right: 301
- simultaneous_music (194)
 - on left: 140 141, on right: 164
- start_symbol (148)
 - on left: 1 3, on right: 0
- steno_duration (279)
 - on left: 405 406, on right: 120 121 407
- steno_pitch (268)
 - on left: 378 379 380, on right: 384
- step_number (301)
 - on left: 461 462 463, on right: 459 460
- step_numbers (300)
 - on left: 459 460, on right: 457 460
- string (244)
 - on left: 302 303 304, on right: 59 304
- string_number_event (261)
 - on left: 352, on right: 364
- sub_quotes (267)
 - on left: 376 377, on right: 372 377 380 383
- sup_quotes (266)
 - on left: 374 375, on right: 373 375 379 382
- tempo_event (185)
 - on left: 120 121 122, on right: 339
- tempo_range (302)
 - on left: 464 465, on right: 120 121
- toplevel_expression (151)
 - on left: 10 11 12 13 14 15 16 17, on right: 5
- tremolo_type (282)
 - on left: 412 413, on right: 366
- unsigned_number (309)

on left: 481 482, on right: 136 137

Annexe D GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its

Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Annexe E Index des commandes LilyPond

Cet index recense toutes les commandes et mots réservés de LilyPond, sous forme de lien vers les endroits du manuel où ils sont abordés. Chaque lien est constitué de deux parties : la première pointe directement là où la commande ou le terme apparaît, la seconde au début de la section où la commande ou le terme est abordé.

!	^
! 5	^ 389
,	-
' 1	- 246
,	\
, 1	\! 112
-	\(..... 122
- 109	\) 122
.	\< 112
. 41	\> 112
/	\\ 156
/ 389	\abs-fontsize 223, 642
/+ 390	\accent 109
:	\accepts 562, 563, 564
: 147	\acciaccatura 102
<	\accidentalStyle 23
< 149	\addChordShape 348
<...> 149	\addInstrumentDefinition 191, 198
=	\addlyrics 240, 242, 243
= 9	\addQuote 192
>	\aeolian 20
> 149	\afterGrace 102
?	\aikenHeads 36
? 5	\aikenHeadsMinor 36
[\alias 562
[..... 86	\allowPageTurn 514
]	\alternative 134
] 86	\appendToTag 477
	\appoggiatura 102
	\arpeggio 129
	\arpeggioArrowDown 129
	\arpeggioArrowUp 129
	\arpeggioBracket 129
	\arpeggioNormal 129
	\arpeggioParenthesis 129
	\arpeggioParenthesisDashed 129
	\arrow-head 230, 665
	\ascendens 423, 430
	\auctum 423, 430
	\augmentum 430
	\auto-footnote 678
	\autoBeamOff 75, 305
	\autoBeamOn 75
	\autochange 303
	\backslashed-digit 679
	\balloonGrobText 207
	\balloonLengthOff 207
	\balloonLengthOn 207
	\balloonText 207
	\bar 90, 94
	\barNumberCheck 99

<code>\beam</code>	665	<code>\doubleflat</code>	671
<code>\bendAfter</code>	124	<code>\doublesharp</code>	671
<code>\bold</code>	223, 642	<code>\downbow</code>	109
<code>\book</code>	446, 449	<code>\downbow</code>	314
<code>\bookOutputName</code>	448	<code>\downmordent</code>	109
<code>\bookOutputSuffix</code>	448	<code>\downprall</code>	109
<code>\bookpart</code>	447, 449, 512	<code>\draw-circle</code>	230, 666
<code>\box</code>	228, 642	<code>\draw-hline</code>	666
<code>\bracket</code>	117, 228, 666	<code>\draw-line</code>	230, 667
<code>\break</code>	512	<code>\drummode</code>	171
<code>\breathe</code>	123	<code>\dynamic</code>	117, 643
<code>\breve</code>	40, 51	<code>\dynamicDown</code>	114
<code>\cadenzaOff</code>	67	<code>\dynamicNeutral</code>	114
<code>\cadenzaOn</code>	67	<code>\dynamicUp</code>	114
<code>\caesura</code>	422	<code>\easyHeadsOff</code>	34
<code>\caps</code>	642	<code>\easyHeadsOn</code>	34
<code>\cavum</code>	423, 430	<code>\epsfile</code>	230, 667
<code>\center-align</code>	225, 650	<code>\espressivo</code>	109, 113
<code>\center-column</code>	227, 651	<code>\expandFullBarRests</code>	55, 56
<code>\change</code>	302	<code>\eyeglasses</code>	679
<code>\char</code>	679	<code>\f</code>	112
<code>\chordmode</code>	12, 345	<code>\featherDurations</code>	89
<code>\chordRepeats</code>	321	<code>\fermata</code>	109
<code>\chords</code>	392	<code>\fermataMarkup</code>	55, 56, 109
<code>\circle</code>	228, 666	<code>\ff</code>	112
<code>\clef</code>	16	<code>\fff</code>	112
<code>\cm</code>	580	<code>\ffff</code>	112
<code>\coda</code>	109	<code>\fffff</code>	112
<code>\column</code>	227, 651	<code>\fill-line</code>	227, 652
<code>\column-lines</code>	685	<code>\fill-with-pattern</code>	653
<code>\combine</code>	230, 651	<code>\filled-box</code>	230, 667
<code>\compoundMeter</code>	71	<code>\finalis</code>	422
<code>\compressFullBarRests</code>	55, 56	<code>\finger</code>	201, 643
<code>\concat</code>	652	<code>\flageolet</code>	109
<code>\consists</code>	562	<code>\flat</code>	671
<code>\context</code>	551, 557	<code>\flexa</code>	430
<code>\cr</code>	112	<code>\fontCaps</code>	643
<code>\cresc</code>	113	<code>\fontsize</code>	223, 643
<code>\crescHairpin</code>	113	<code>\footnote</code>	464, 679
<code>\crescTextCresc</code>	113	<code>\fp</code>	112
<code>\crossStaff</code>	305	<code>\fraction</code>	679
<code>\cueClef</code>	195	<code>\frenchChords</code>	396
<code>\cueDuring</code>	195	<code>\fret-diagram</code>	336, 675
<code>\cueDuringWithClef</code>	195	<code>\fret-diagram-terse</code>	338, 676
<code>\customTabClef</code>	671	<code>\fret-diagram-verbose</code>	339, 676
<code>\decr</code>	112	<code>\fromproperty</code>	680
<code>\decresc</code>	113	<code>\funkHeads</code>	36
<code>\defaultTimeSignature</code>	59	<code>\funkHeadsMinor</code>	36
<code>\deminutum</code>	423, 430	<code>\general-align</code>	226, 654
<code>\denies</code>	562, 563, 564	<code>\germanChords</code>	396
<code>\descendens</code>	423, 430	<code>\glissando</code>	125
<code>\dim</code>	113	<code>\grace</code>	102
<code>\dimHairpin</code>	113	<code>\halfopen</code>	109
<code>\dimTextDecr</code>	113	<code>\halign</code>	225, 655
<code>\dimTextDecresc</code>	113	<code>\harmonic</code>	315, 324
<code>\dimTextDim</code>	113	<code>\harmonicByFret</code>	324
<code>\dir-column</code>	652	<code>\harmonicByRatio</code>	324
<code>\displayLilyMusic</code>	494	<code>\harmonicsOff</code>	315
<code>\divisioMaior</code>	422	<code>\harmonicsOn</code>	315
<code>\divisioMaxima</code>	422	<code>\harp-pedal</code>	677
<code>\divisioMinima</code>	422	<code>\hbracket</code>	228, 668
<code>\dorian</code>	20	<code>\hcenter-in</code>	656
<code>\dotsDown</code>	41	<code>\header</code>	449
<code>\dotsNeutral</code>	41	<code>\hideKeySignature</code>	375
<code>\dotsUp</code>	41	<code>\hideNotes</code>	204

<code>\hideSplitTiedTabNotes</code>	323	<code>\modalInversion</code>	15
<code>\hideStaffSwitch</code>	305	<code>\modalTranspose</code>	14
<code>\hspace</code>	656	<code>\mordent</code>	109
<code>\huge</code>	200, 225, 643	<code>\mp</code>	112
<code>\improvisationOff</code>	39, 73	<code>\musicglyph</code>	101, 672
<code>\improvisationOn</code>	39, 73	<code>\name</code>	562
<code>\in</code>	580	<code>\natural</code>	672
<code>\inclinatum</code>	423, 430	<code>\new</code>	551
<code>\include</code>	473	<code>\noBeam</code>	87
<code>\inStaffSegno</code>	137	<code>\noBreak</code>	512
<code>\instrumentSwitch</code>	191	<code>\noPageBreak</code>	513
<code>\inversion</code>	13	<code>\noPageTurn</code>	514
<code>\ionian</code>	20	<code>\normal-size-sub</code>	645
<code>\italianChords</code>	396	<code>\normal-size-super</code>	224, 645
<code>\italic</code>	223, 644	<code>\normal-text</code>	646
<code>\justified-lines</code>	233, 685	<code>\normalsize</code>	200, 225, 646
<code>\justify</code>	227, 657	<code>\note</code>	672
<code>\justify-field</code>	657	<code>\note-by-number</code>	672
<code>\justify-string</code>	658	<code>\null</code>	226, 681
<code>\keepWithTag</code>	477	<code>\number</code>	646
<code>\key</code>	19, 36	<code>\numericTimeSignature</code>	59
<code>\killCues</code>	198	<code>\octaveCheck</code>	9
<code>\label</code>	471	<code>\on-the-fly</code>	462, 681
<code>\laissezVibrer</code>	48	<code>\once</code>	573
<code>\large</code>	200, 225, 644	<code>\oneVoice</code>	155
<code>\larger</code>	223, 225, 644	<code>\open</code>	109
<code>\layout</code>	449, 507, 557	<code>\open</code>	314
<code>\left-align</code>	225, 658	<code>\oriscus</code>	423, 430
<code>\left-brace</code>	680	<code>\ottava</code>	21
<code>\left-column</code>	659	<code>\override</code>	571, 681
<code>\lheel</code>	109	<code>\override en mode \lyricmode</code>	239
<code>\line</code>	659	<code>\override-lines</code>	685
<code>\linea</code>	423, 430	<code>\overrideTimeSignatureSettings</code>	59
<code>\lineprall</code>	109	<code>\p</code>	112
<code>\locrian</code>	20	<code>\pad-around</code>	229, 659
<code>\longa</code>	40, 51	<code>\pad-markup</code>	229, 660
<code>\longfermata</code>	109	<code>\pad-to-box</code>	229, 660
<code>\lookup</code>	680	<code>\pad-x</code>	229, 660
<code>\lower</code>	226, 659	<code>\page-link</code>	682
<code>\ltoe</code>	109	<code>\page-ref</code>	471, 682
<code>\lydian</code>	20	<code>\pageBreak</code>	513
<code>\lyricmode</code>	239, 240	<code>\pageTurn</code>	514
<code>\lyricsto</code>	240, 242, 243	<code>\paper</code>	449, 497
<code>\magnify</code>	223, 644	<code>\parallelMusic</code>	168
<code>\major</code>	20	<code>\parenthesize</code>	206, 668
<code>\makeClusters</code>	154	<code>\partcombine</code>	163
<code>\marcato</code>	109	<code>\partcombineApart</code>	164
<code>\mark</code>	100, 216	<code>\partcombineAutomatic</code>	164
<code>\markalphabet</code>	681	<code>\partcombineChords</code>	164
<code>\markletter</code>	681	<code>\partcombineSoloI</code>	164
<code>\markup</code>	216, 219, 220, 221	<code>\partcombineSoloII</code>	164
<code>\markuplist</code>	220, 233	<code>\partcombineUnisono</code>	164
<code>\maxima</code>	40, 51	<code>\partial</code>	66, 134, 136
<code>\medium</code>	645	<code>\path</code>	669
<code>\melisma</code>	247	<code>\pattern</code>	682
<code>\melismaEnd</code>	247	<code>\pes</code>	430
<code>\mergeDifferentlyDottedOff</code>	158	<code>\phrasingSlurDashed</code>	122
<code>\mergeDifferentlyDottedOn</code>	158	<code>\phrasingSlurDashPattern</code>	122
<code>\mergeDifferentlyHeadedOff</code>	158	<code>\phrasingSlurDotted</code>	122
<code>\mergeDifferentlyHeadedOn</code>	158	<code>\phrasingSlurDown</code>	122
<code>\mf</code>	112	<code>\phrasingSlurHalfDashed</code>	122
<code>\midi</code>	449	<code>\phrasingSlurHalfSolid</code>	122
<code>\minor</code>	20	<code>\phrasingSlurNeutral</code>	122
<code>\mixolydian</code>	20	<code>\phrasingSlurSolid</code>	122
<code>\mm</code>	580	<code>\phrasingSlurUp</code>	122

<code>\phrygian</code>	20	<code>\shiftOff</code>	158
<code>\pitchedTrill</code>	133	<code>\shiftOn</code>	158
<code>\portato</code>	109	<code>\shiftOnn</code>	158
<code>\postscript</code>	230, 669	<code>\shiftOnnn</code>	158
<code>\powerChords</code>	360	<code>\shortfermata</code>	109
<code>\pp</code>	112	<code>\showKeySignature</code>	375
<code>\ppp</code>	112	<code>\showStaffSwitch</code>	305
<code>\pppp</code>	112	<code>\signumcongruentiae</code>	109
<code>\ppppp</code>	112	<code>\simple</code>	647
<code>\prall</code>	109	<code>\skip</code>	53
<code>\pralldown</code>	109	<code>\slashed-digit</code>	683
<code>\prallmordent</code>	109	<code>\slashedGrace</code>	102
<code>\prallprall</code>	109	<code>\slurDashed</code>	119
<code>\prallup</code>	109	<code>\slurDashPattern</code>	120
<code>\predefinedFretboardsOff</code>	355	<code>\slurDotted</code>	119
<code>\predefinedFretboardsOn</code>	355	<code>\slurDown</code>	119
<code>\property-recursive</code>	682	<code>\slurHalfDashed</code>	120
<code>\pt</code>	580	<code>\slurHalfSolid</code>	120
<code>\pushToTag</code>	477	<code>\slurNeutral</code>	119
<code>\put-adjacent</code>	661	<code>\slurSolid</code>	119
<code>\quilisma</code>	423, 430	<code>\slurUp</code>	120
<code>\quoteDuring</code>	192, 195	<code>\small</code>	200, 225, 647
<code>\raise</code>	226, 661	<code>\smallCaps</code>	648
<code>\relative</code>	2, 12, 304	<code>\smaller</code>	223, 225, 648
<code>\RemoveEmptyStaves</code>	185	<code>\snappizzicato</code>	109
<code>\RemoveEmptyStaves</code>	186	<code>\sostenutoOff</code>	307
<code>\removeWithTag</code>	477	<code>\sostenutoOn</code>	307
<code>\repeat</code>	134	<code>\southernHarmonyHeads</code>	36
<code>\repeat percent</code>	145	<code>\southernHarmonyHeadsMinor</code>	36
<code>\repeat tremolo</code>	147	<code>\sp</code>	112
<code>\repeat unfold</code>	143	<code>\spp</code>	112
<code>\repeatTie</code>	48, 137, 264	<code>\staccatissimo</code>	109
<code>\replace</code>	646	<code>\staccato</code>	109
<code>\rest</code>	51	<code>\startGroup</code>	211
<code>\retrograde</code>	13	<code>\startStaff</code>	178, 181
<code>\reverseturn</code>	109	<code>\startTrillSpan</code>	132
<code>\revert</code>	572	<code>\stemDown</code>	207
<code>\revertTimeSignatureSettings</code>	61	<code>\stemNeutral</code>	207
<code>\rfz</code>	112	<code>\stemUp</code>	207
<code>\rheel</code>	109	<code>\stencil</code>	683
<code>\right-align</code>	225, 661	<code>\stopGroup</code>	211
<code>\right-brace</code>	683	<code>\stopped</code>	109
<code>\right-column</code>	661	<code>\stopStaff</code>	178, 181, 185
<code>\rightHandFinger</code>	357	<code>\stopTrillSpan</code>	132
<code>\roman</code>	647	<code>\storePredefinedDiagram</code>	348
<code>\rotate</code>	662	<code>\stringTuning</code>	332
<code>\rounded-box</code>	228, 670	<code>\strophae</code>	423, 430
<code>\rtoe</code>	109	<code>\strut</code>	683
<code>\sacredHarpHeads</code>	36	<code>\sub</code>	224, 648
<code>\sacredHarpHeadsMinor</code>	36	<code>\super</code>	224, 648
<code>\sans</code>	647	<code>\sustainOff</code>	307
<code>\scale</code>	670	<code>\sustainOn</code>	307
<code>\scaleDurations</code>	47, 69	<code>\tabChordRepeats</code>	321
<code>\score</code>	445, 449, 673	<code>\tabFullNotation</code>	320
<code>\segno</code>	109	<code>\table-of-contents</code>	473, 685
<code>\semiflat</code>	674	<code>\tag</code>	477
<code>\semiGermanChords</code>	396	<code>\taor</code>	375
<code>\semisharp</code>	674	<code>\teeny</code>	200, 225, 649
<code>\sesquiflat</code>	674	<code>\tempo</code>	63
<code>\sesquisharp</code>	674	<code>\tenuto</code>	109
<code>\set</code>	78, 569	<code>\text</code>	649
<code>\sf</code>	112	<code>\textLengthOff</code>	56, 213
<code>\sff</code>	112	<code>\textLengthOn</code>	56, 213
<code>\sfz</code>	112	<code>\textSpannerDown</code>	214
<code>\sharp</code>	674	<code>\textSpannerNeutral</code>	214

<code>\textSpannerUp</code>	214
<code>\thumb</code>	109, 202
<code>\tied-lyric</code>	675
<code>\tieDashed</code>	48
<code>\tieDotted</code>	48
<code>\tieDown</code>	48
<code>\tieNeutral</code>	48
<code>\tieSolid</code>	48
<code>\tieUp</code>	48
<code>\time</code>	58, 78
<code>\times</code>	42, 69
<code>\tiny</code>	200, 225, 649
<code>\tocItem</code>	473
<code>\translate</code>	226, 662
<code>\translate-scaled</code>	226, 662
<code>\transparent</code>	683
<code>\transpose</code>	10, 12
<code>\transposedCueDuring</code>	198
<code>\transposition</code>	22, 192
<code>\treCorde</code>	307
<code>\triangle</code>	230, 671
<code>\trill</code>	109, 132
<code>\tupletDown</code>	42
<code>\tupletNeutral</code>	42
<code>\tupletUp</code>	42
<code>\turn</code>	109
<code>\tweak</code>	573
<code>\type</code>	562
<code>\typewriter</code>	649
<code>\unaCorda</code>	307
<code>\underline</code>	223, 650
<code>\unfoldRepeats</code>	489
<code>\unHideNotes</code>	204
<code>\unset</code>	570
<code>\upbow</code>	109
<code>\upbow</code>	314
<code>\upmordent</code>	109
<code>\upprall</code>	109
<code>\upright</code>	650
<code>\varcoda</code>	109
<code>\vcenter</code>	663
<code>\verbatim-file</code>	683
<code>\verylongfermata</code>	109
<code>\virga</code>	423, 430
<code>\virgula</code>	422
<code>\voiceFourStyle</code>	158
<code>\voiceNeutralStyle</code>	158
<code>\voiceOne</code>	155
<code>\voiceOne ... \voiceFour</code>	155
<code>\voiceOneStyle</code>	158
<code>\voiceThreeStyle</code>	158
<code>\voiceTwoStyle</code>	158
<code>\void</code>	495
<code>\vspace</code>	663
<code>\walkerHeads</code>	36
<code>\walkerHeadsMinor</code>	36
<code>\whiteout</code>	684
<code>\with</code>	555, 560
<code>\with-color</code>	205, 684
<code>\with-dimensions</code>	684
<code>\with-link</code>	684
<code>\with-url</code>	671
<code>\woodwind-diagram</code>	677
<code>\wordwrap</code>	227, 664
<code>\wordwrap-field</code>	663

<code>\wordwrap-internal</code>	685
<code>\wordwrap-lines</code>	233, 685
<code>\wordwrap-string</code>	664
<code>\wordwrap-string-internal</code>	685

|

.....	99
-------	----

~

~	47
---------	----

A

<code>accepts</code>	562
<code>acciaccatura</code>	722
<code>accidentalStyle</code>	722
<code>addChordShape</code>	348
<code>addChordShape</code>	722
<code>addInstrumentDefinition</code>	191, 198, 722
<code>additionalPitchPrefix</code>	394
<code>addQuote</code>	192, 722
<code>aeolian</code>	20
<code>afterGrace</code>	102, 722
<code>aikenHeads</code>	36
<code>aikenHeadsMinor</code>	36
<code>alias</code>	562
<code>alignAboveContext</code>	564
<code>alignBelowContext</code>	261, 564
<code>allowPageTurn</code>	722
<code>alterBroken</code>	722
<code>annotate-spacing</code>	546
<code>appendToTag</code>	722
<code>applyContext</code>	722
<code>applyMusic</code>	722
<code>applyOutput</code>	722
<code>appoggiatura</code>	723
<code>arpeggio</code>	129
<code>arpeggioArrowDown</code>	129
<code>arpeggioArrowUp</code>	129
<code>arpeggioBracket</code>	129
<code>arpeggioNormal</code>	129
<code>arpeggioParenthesis</code>	129
<code>arpeggioParenthesisDashed</code>	129
<code>arrow-head</code>	230
<code>assertBeamQuant</code>	723
<code>assertBeamSlope</code>	723
<code>aug</code>	387
<code>auto-first-page-number</code>	505
<code>autoBeaming</code>	78
<code>autoBeamOff</code>	75
<code>autoBeamOn</code>	75
<code>autochange</code>	303, 723

B

<code>Balloon_engraver</code>	207
<code>balloonGrobText</code>	207, 723
<code>balloonLengthOff</code>	207
<code>balloonLengthOn</code>	207
<code>balloonText</code>	207, 723
<code>banjo-c-tuning</code>	362
<code>banjo-modal-tuning</code>	362

banjo-open-d-tuning..... 362
 banjo-open-dm-tuning..... 362
 bar..... 90, 94, 723
 barCheckSynchronize..... 99
 BarNumber..... 95
 barNumberCheck..... 99, 723
 barNumberVisibility..... 95
 bartype..... 94
 base-shortest-duration..... 534
 baseMoment..... 78
 beamExceptions..... 78
 beatStructure..... 78
 bendAfter..... 124, 723
 binding-offset..... 503
 blank-after-score-page-force..... 504
 blank-last-page-force..... 504
 blank-page-force..... 505
 bold..... 223
 bookOutputName..... 723
 bookOutputSuffix..... 723
 bookTitleMarkup..... 459
 bottom-margin..... 499
 box..... 228
 bracket..... 117, 228, 308
 breakable..... 76
 breathe..... 123
 breathe..... 723
 breve..... 40, 51

C

cadenzaOff..... 67
 cadenzaOn..... 67
 center-align..... 225
 center-column..... 227
 change..... 302
 check-consistency..... 502
 chordChanges..... 392
 chordmode..... 12, 345
 chordNameExceptions..... 395
 chordNameLowercaseMinor..... 394
 ChordNames..... 345
 chordNameSeparator..... 394
 chordNoteNamer..... 394
 chordPrefixSpacer..... 396
 chordRepeats..... 723
 chordRootNamer..... 394
 circle..... 228
 clef..... 16, 723
 color..... 205
 column..... 227
 combine..... 230
 common-shortest-duration..... 534
 Completion_heads_engraver..... 72
 Completion_rest_engraver..... 72
 compoundMeter..... 723
 compressFullBarRests..... 55, 56
 consists..... 562
 controlpitch..... 9
 cr..... 112
 cresc..... 113
 crescHairpin..... 113
 crescTextCresc..... 113
 cross..... 33

crossStaff..... 723
 cueClef..... 195, 723
 cueClefUnset..... 723
 cueDuring..... 195, 724
 cueDuringWithClef..... 195, 724
 currentBarNumber..... 94
 currentBarNumber..... 108

D

deadNote..... 724
 decr..... 112
 decresc..... 113
 default..... 23, 25
 defaultBarType..... 94
 defaultNoteHeads..... 724
 defaultTimeSignature..... 59
 denies..... 562
 dim..... 113, 387
 dimHairpin..... 113
 dimTextDecr..... 113
 dimTextDecresc..... 113
 dimTextDim..... 113
 displayLilyMusic..... 724
 displayMusic..... 724
 dodecaphonic..... 28
 dorian..... 20
 dotsDown..... 41
 dotsNeutral..... 41
 dotsUp..... 41
 draw-circle..... 230
 draw-line..... 230
 drummode..... 171
 DrumStaff..... 171
 dynamic..... 117
 dynamicDown..... 114
 DynamicLineSpanner..... 114
 dynamicNeutral..... 114
 dynamicUp..... 114

E

easyHeadsOff..... 34
 easyHeadsOn..... 34
 endSpanners..... 724
 epsfile..... 230
 espressivo..... 113
 eventChords..... 724
 expandFullBarRests..... 55, 56

F

f..... 112
 featherDurations..... 89, 724
 fermataMarkup..... 55, 56
 ff..... 112
 fff..... 112
 ffff..... 112
 fffff..... 112
 fill-line..... 227
 filled-box..... 230
 finger..... 201
 first-page-number..... 505
 followVoice..... 305

font-interface 201, 234
font-size 200, 201
fontsize 223
fontSize 200
footnote 724
forget 29
four-string-banjo 362
fp 112
fret-diagram 336
fret-diagram-interface 341
fret-diagram-terse 338
fret-diagram-verbose 339
FretBoards 344
funkHeads 36
funkHeadsMinor 36

G

general-align 226
glissando 125
grace 724
GregorianTranscriptionStaff 171
Grid_line_span_engraver 208
Grid_point_engraver 208
gridInterval 208
grobdescriptions 724
grow-direction 89

H

halign 225
harmonicByFret 724
harmonicByRatio 724
harmonicNote 725
harmonicsOn 725
hbracket 228
hideKeySignature 375
hideNotes 204
hideStaffSwitch 305
horizontal-shift 504
Horizontal_bracket_engraver 211
huge 200, 225

I

improvisationOff 39, 73
improvisationOn 39, 73
indent 189, 504, 537
inner-margin 503
inStaffSegno 725
instrumentSwitch 191, 725
inversion 725
ionian 20
italic 223

J

justified-lines 233
justify 227

K

keepWithTag 725
key 19, 36, 725

killCues 198, 725

L

label 725
laissezVibrer 48
language 725
languageRestore 725
languageSaveAndChange 725
large 200, 225
larger 223, 225
last-bottom-spacing 501
layout file 509
left-align 225
left-margin 502
line-width 502, 537
locrian 20
longa 40, 51
lower 226
ly:minimal-breaking 514
ly:one-line-breaking 514
ly:optimal-breaking 513
ly:page-turn-breaking 513
lydian 20

M

m 387
magnify 223
magstep 200, 580
maj 387
major 20
majorSevenSymbol 394
make-dynamic-script 118
make-pango-font-tree 236
makeClusters 154, 725
makeDefaultStringTuning 725
mark 100, 216, 725
markup 216, 219, 220, 221
markup-markup-spacing 501
markup-system-spacing 501
markuplist 220, 233
max-systems-per-page 504
maxima 40, 51
measureLength 78
measureLength 108
measurePosition 66
measurePosition 108
MensuralStaff 171
mergeDifferentlyDottedOff 158
mergeDifferentlyDottedOn 158
mergeDifferentlyHeadedOff 158
mergeDifferentlyHeadedOn 158
mf 112
min-systems-per-page 504
minimumFret 320
minimumPageTurnLength 513
minimumRepeatLengthForPageTurn 514
minor 20
minorChordModifier 395
mixed 308
mixolydian 20
modalInversion 15, 725
modalTranspose 14, 725

modern	26
modern-cautionary	26
modern-voice	26
modern-voice-cautionary	26
mp	112
MultiMeasureRestText	55
musicglyph	101
musicMap	725

N

name	562
neo-modern	27
neo-modern-cautionary	27
neo-modern-voice	28
neo-modern-voice-cautionary	28
no-reset	29
noBeam	87
noPageBreak	725
noPageTurn	725
normal-size-super	224
normalsize	200, 225
Note_heads_engraver	72
null	226
numericTimeSignature	59

O

octaveCheck	9, 726
once	726
oneVoice	155
ottava	21, 726
outer-margin	503
outside-staff-horizontal-padding	532
outside-staff-padding	532
outside-staff-priority	532
overrideProperty	726
overrideTimeSignatureSettings	726

P

p	112
pad-around	229
pad-markup	229
pad-to-box	229
pad-x	229
page-breaking	505
page-breaking-system-system-spacing	505
page-count	505
page-spacing-weight	506
pageBreak	726
pageTurn	726
palmMute	726
palmMuteOn	726
paper-height	499
paper-width	502
parallelMusic	168, 726
parenthesize	206, 726
partcombine	163, 726
partcombineApart	164
partcombineAutomatic	164
partcombineChords	164
partcombineDown	726
partcombineForce	726

partcombineSoloI	164
partcombineSoloII	164
partcombineUnisono	164
partcombineUp	727
partial	66, 727
pedalSustainStyle	308
percent	145
phrasingSlurDashed	122
phrasingSlurDashPattern	122, 727
phrasingSlurDotted	122
phrasingSlurDown	122
phrasingSlurHalfDashed	122
phrasingSlurHalfSolid	122
phrasingSlurNeutral	122
phrasingSlurSolid	122
phrasingSlurUp	122
phrygian	20
piano	27
piano-cautionary	27
PianoStaff	301, 303
pipeSymbol	99
Pitch_squash_engraver	73
pitchedTrill	133, 727
pointAndClickOff	727
pointAndClickOn	727
pointAndClickTypes	727
postscript	230
powerChords	360
pp	112
ppp	112
pppp	112
ppppp	112
predefinedFretboardsOff	355
predefinedFretboardsOn	355
print-all-headers	506
print-first-page-number	505
print-page-number	505
pushToTag	727

Q

quotedCueEventTypes	194
quotedEventTypes	194
quoteDuring	192, 195, 727

R

r	51
R	54
ragged-bottom	499
ragged-last	503, 537
ragged-last-bottom	499
ragged-right	502, 537
raise	226
relative	2, 12, 304, 727
removeWithTag	727
repeatCommands	140
repeatTie	48
resetRelativeOctave	727
rest	51
restrainOpenStrings	320
retrograde	13, 727
revertTimeSignatureSettings	727
rfz	112

rgb-color	205
RhythmicStaff	171
right-align	225
right-margin	502
rightHandFinger	357, 727
rounded-box	228

S

s	53
sacredHarpHeads	36
sacredHarpHeadsMinor	36
scaleDurations	47, 69, 727
score-markup-spacing	501
score-system-spacing	501
scoreTitleMarkup	459
set	78
set-octavation	21
settingsFrom	727
sf	112
sff	112
sfz	112
shape	728
shiftDurations	728
shiftOff	158
shiftOn	158
shiftOnn	158
shiftOnnn	158
short-indent	189, 504
show-available-fonts	236
showFirstLength	483
showKeySignature	375
showLastLength	483
showStaffSwitch	305
skip	53, 728
skipTypesetting	483
slashChordSeparator	395
slashedGrace	728
slurDashed	119
slurDashPattern	120, 728
slurDotted	119
slurDown	119
slurHalfDashed	120
slurHalfSolid	120
slurNeutral	119
slurSolid	119
slurUp	120
small	200, 225
smaller	223, 225
sostenutoOff	307
sostenutoOn	307
southernHarmonyHeads	36
southernHarmonyHeadsMinor	36
sp	112
spacing	534
spacingTweaks	728
Span_stem_engraver	305
spp	112
Staff.midiInstrument	486
Staff_symbol_engraver	185
start-repeat	140
startGroup	211
startStaff	178, 181
startTrillSpan	132
Stem	305

stem-spacing-correction	534
stemDown	207
stemLeftBeamCount	87
stemNeutral	207
stemRightBeamCount	87
stemUp	207
stopGroup	211
stopStaff	178, 181, 185
stopTrillSpan	132
storePredefinedDiagram	348
storePredefinedDiagram	728
stringTuning	332, 728
stringTunings	331, 344
styledNoteHeads	728
sub	224
suggestAccidentals	417
super	224
sus	389
sustainOff	307
sustainOn	307
symboles de septième majeure	396
system-count	504
system-separator-markup	506
system-system-spacing	501
systems-per-page	504

T

tabChordRepeats	728
tabChordRepetition	728
TabStaff	171, 319
TabVoice	319
tag	728
taor	375
teaching	28
teeny	200, 225
tempo	63
text	308
textLengthOff	213
textLengthOn	56, 213
textLenthOff	56
textSpannerDown	214
textSpannerNeutral	214
textSpannerUp	214
thumb	202
tieDashed	48
tieDashPattern	728
tieDotted	48
tieDown	48
tieNeutral	48
tieSolid	48
tieUp	48
time	58, 78, 728
times	42, 69, 728
timeSignatureFraction	69
tiny	200, 225
tocItem	728
top-margin	499
top-markup-spacing	501
top-system-spacing	501
translate	226
translate-scaled	226
transpose	10, 12, 729
transposedCueDuring	198, 729

transposition.....	22, 192, 729
treCorde.....	307
tremolo.....	147
tremoloFlags.....	147
triangle.....	230
trill.....	132
tupletDown.....	42
tupletNeutral.....	42
TupletNumber.....	43
tupletNumberFormatFunction.....	43
tupletSpannerDuration.....	43
tupletUp.....	42
tweak.....	729
two-sided.....	503
type.....	562

U

unaCorda.....	307
underline.....	223
unfold.....	143
unfoldRepeats.....	729
unHideNotes.....	204

V

VaticanaStaff.....	171
voice.....	23, 25
Voice.....	155
voiceOne.....	155
void.....	729

W

walkerHeads.....	36
walkerHeadsMinor.....	36
whichBar.....	94
with-color.....	205
withMusicProperty.....	729
wordwrap.....	227
wordwrap-lines.....	233

X

x11-color.....	205, 206
xNote.....	729
xNotesOn.....	729

Annexe F Index de LilyPond

En plus des commandes et mots réservés de LilyPond, cet index recense les termes musicaux qui s'y rapportent. Tout comme dans l'index des commandes, chaque lien est constitué de deux parties : la première pointe directement là où la commande ou le terme apparaît, la seconde au début de la section où la commande ou le terme est abordé.

!	^
!..... 5	^..... 389
,	-
'..... 1	-..... 246
,	\
,..... 1	\!..... 112
-	\(..... 122
-..... 109	\)..... 122
.	\<..... 112
..... 41	\>..... 112
/	\\..... 156
/..... 389	\abs-fontsize..... 223, 642
/+..... 390	\accent..... 109
:	\accepts..... 562, 563, 564
:..... 147	\acciaccatura..... 102
<	\accidentalStyle..... 23
<..... 149	\addChordShape..... 348
<...>..... 149	\addInstrumentDefinition..... 191, 198
=	\addlyrics..... 240, 242, 243
=..... 9	\addQuote..... 192
>	\aeolian..... 20
>..... 149	\afterGrace..... 102
?	\aikenHeads..... 36
?..... 5	\aikenHeadsMinor..... 36
[\alias..... 562
[..... 86	\allowPageTurn..... 514
]	\alternative..... 134
]..... 86	\appendToTag..... 477
	\appoggiatura..... 102
	\arpeggio..... 129
	\arpeggioArrowDown..... 129
	\arpeggioArrowUp..... 129
	\arpeggioBracket..... 129
	\arpeggioNormal..... 129
	\arpeggioParenthesis..... 129
	\arpeggioParenthesisDashed..... 129
	\arrow-head..... 230, 665
	\ascendens..... 423, 430
	\auctum..... 423, 430
	\augmentum..... 430
	\auto-footnote..... 678
	\autoBeamOff..... 75, 305
	\autoBeamOn..... 75
	\autochange..... 303
	\backslashed-digit..... 679
	\balloonGrobText..... 207
	\balloonLengthOff..... 207
	\balloonLengthOn..... 207
	\balloonText..... 207
	\bar..... 90, 94
	\barNumberCheck..... 99

<code>\beam</code>	665	<code>\dotsUp</code>	41
<code>\bendAfter</code>	124	<code>\doubleflat</code>	671
<code>\bold</code>	223, 642	<code>\doublesharp</code>	671
<code>\book</code>	446, 449	<code>\downbow</code>	109
<code>\bookOutputName</code>	448	<code>\downbow</code>	314
<code>\bookOutputSuffix</code>	448	<code>\downmordent</code>	109
<code>\bookpart</code>	447, 449, 512	<code>\downprall</code>	109
<code>\box</code>	228, 642	<code>\draw-circle</code>	230, 666
<code>\bracket</code>	117, 228, 666	<code>\draw-hline</code>	666
<code>\break</code>	512	<code>\draw-line</code>	230, 667
<code>\breathe</code>	123	<code>\drummode</code>	171
<code>\breve</code>	40, 51	<code>\dynamic</code>	117, 643
<code>\cadenzaOff</code>	67	<code>\dynamicDown</code>	114
<code>\cadenzaOn</code>	67	<code>\dynamicNeutral</code>	114
<code>\caesura</code>	422	<code>\dynamicUp</code>	114
<code>\caps</code>	642	<code>\easyHeadsOff</code>	34
<code>\cavum</code>	423, 430	<code>\easyHeadsOn</code>	34
<code>\center-align</code>	225, 650	<code>\epsfile</code>	230, 667
<code>\center-column</code>	227, 651	<code>\espressivo</code>	109, 113
<code>\change</code>	302	<code>\expandFullBarRests</code>	55, 56
<code>\char</code>	679	<code>\eyeglasses</code>	679
<code>\chordmode</code>	12, 345	<code>\f</code>	112
<code>\chordRepeats</code>	321	<code>\featherDurations</code>	89
<code>\chords</code>	392	<code>\fermata</code>	109
<code>\circle</code>	228, 666	<code>\fermataMarkup</code>	55, 56, 109
<code>\clef</code>	16	<code>\ff</code>	112
<code>\cm</code>	580	<code>\fff</code>	112
<code>\coda</code>	109	<code>\ffff</code>	112
<code>\column</code>	227, 651	<code>\fffff</code>	112
<code>\column-lines</code>	685	<code>\fill-line</code>	227, 652
<code>\combine</code>	230, 651	<code>\fill-with-pattern</code>	653
<code>\compoundMeter</code>	71	<code>\filled-box</code>	230, 667
<code>\compressFullBarRests</code>	55, 56	<code>\finalis</code>	422
<code>\concat</code>	652	<code>\finger</code>	201, 643
<code>\consists</code>	562	<code>\flageolet</code>	109
<code>\context</code>	551, 557	<code>\flat</code>	671
<code>\context dans un bloc \layout</code>	557	<code>\flexa</code>	430
<code>\cr</code>	112	<code>\fontCaps</code>	643
<code>\cresc</code>	113	<code>\fontsize</code>	223, 643
<code>\crescHairpin</code>	113	<code>\footnote</code>	464, 679
<code>\crescTextCresc</code>	113	<code>\fp</code>	112
<code>\crossStaff</code>	305	<code>\fraction</code>	679
<code>\cueClef</code>	195	<code>\frenchChords</code>	396
<code>\cueDuring</code>	195	<code>\fret-diagram</code>	336, 675
<code>\cueDuringWithClef</code>	195	<code>\fret-diagram-terse</code>	338, 676
<code>\customTabClef</code>	671	<code>\fret-diagram-verbose</code>	339, 676
<code>\decr</code>	112	<code>\fromproperty</code>	680
<code>\decresc</code>	113	<code>\funkHeads</code>	36
<code>\defaultTimeSignature</code>	59	<code>\funkHeadsMinor</code>	36
<code>\deminutum</code>	423, 430	<code>\general-align</code>	226, 654
<code>\denies</code>	562, 563, 564	<code>\germanChords</code>	396
<code>\descendens</code>	423, 430	<code>\glissando</code>	125
<code>\dim</code>	113	<code>\grace</code>	102
<code>\dimHairpin</code>	113	<code>\halfopen</code>	109
<code>\dimTextDecr</code>	113	<code>\halign</code>	225, 655
<code>\dimTextDecresc</code>	113	<code>\harmonic</code>	315, 324
<code>\dimTextDim</code>	113	<code>\harmonicByFret</code>	324
<code>\dir-column</code>	652	<code>\harmonicByRatio</code>	324
<code>\displayLilyMusic</code>	494	<code>\harmonicsOff</code>	315
<code>\divisioMaior</code>	422	<code>\harmonicsOn</code>	315
<code>\divisioMaxima</code>	422	<code>\harp-pedal</code>	677
<code>\divisioMinima</code>	422	<code>\hbracket</code>	228, 668
<code>\dorian</code>	20	<code>\hcenter-in</code>	656
<code>\dotsDown</code>	41	<code>\header</code>	449
<code>\dotsNeutral</code>	41	<code>\hideKeySignature</code>	375

<code>\hideNotes</code>	204	<code>\mm</code>	580
<code>\hideSplitTiedTabNotes</code>	323	<code>\modalInversion</code>	15
<code>\hideStaffSwitch</code>	305	<code>\modalTranspose</code>	14
<code>\hspace</code>	656	<code>\mordent</code>	109
<code>\huge</code>	200, 225, 643	<code>\mp</code>	112
<code>\improvisationOff</code>	39, 73	<code>\musicglyph</code>	101, 672
<code>\improvisationOn</code>	39, 73	<code>\name</code>	562
<code>\in</code>	580	<code>\natural</code>	672
<code>\inclinatum</code>	423, 430	<code>\new</code>	551
<code>\include</code>	473	<code>\noBeam</code>	87
<code>\inStaffSegno</code>	137	<code>\noBreak</code>	512
<code>\instrumentSwitch</code>	191	<code>\noPageBreak</code>	513
<code>\inversion</code>	13	<code>\noPageTurn</code>	514
<code>\ionian</code>	20	<code>\normal-size-sub</code>	645
<code>\italianChords</code>	396	<code>\normal-size-super</code>	224, 645
<code>\italic</code>	223, 644	<code>\normal-text</code>	646
<code>\justified-lines</code>	233, 685	<code>\normalsize</code>	200, 225, 646
<code>\justify</code>	227, 657	<code>\note</code>	672
<code>\justify-field</code>	657	<code>\note-by-number</code>	672
<code>\justify-string</code>	658	<code>\null</code>	226, 681
<code>\keepWithTag</code>	477	<code>\number</code>	646
<code>\key</code>	19, 36	<code>\numericTimeSignature</code>	59
<code>\killCues</code>	198	<code>\octaveCheck</code>	9
<code>\label</code>	471	<code>\on-the-fly</code>	462, 681
<code>\laissezVibrer</code>	48	<code>\once</code>	571
<code>\large</code>	200, 225, 644	<code>\once</code>	573
<code>\larger</code>	223, 225, 644	<code>\oneVoice</code>	155
<code>\layout</code>	449, 507, 557	<code>\open</code>	109
<code>\left-align</code>	225, 658	<code>\open</code>	314
<code>\left-brace</code>	680	<code>\oriscus</code>	423, 430
<code>\left-column</code>	659	<code>\ottava</code>	21
<code>\lheel</code>	109	<code>\override</code>	571, 681
<code>\line</code>	659	<code>\override en mode \lyricmode</code>	239
<code>\linea</code>	423, 430	<code>\override-lines</code>	685
<code>\lineprall</code>	109	<code>\overrideTimeSignatureSettings</code>	59
<code>\locrian</code>	20	<code>\p</code>	112
<code>\longa</code>	40, 51	<code>\pad-around</code>	229, 659
<code>\longfermata</code>	109	<code>\pad-markup</code>	229, 660
<code>\lookup</code>	680	<code>\pad-to-box</code>	229, 660
<code>\lower</code>	226, 659	<code>\pad-x</code>	229, 660
<code>\ltoe</code>	109	<code>\page-link</code>	682
<code>\lydian</code>	20	<code>\page-ref</code>	471, 682
<code>\lyricmode</code>	239, 240	<code>\pageBreak</code>	513
<code>\lyricsto</code>	240, 242, 243	<code>\pageTurn</code>	514
<code>\magnify</code>	223, 644	<code>\paper</code>	449, 497
<code>\major</code>	20	<code>\parallelMusic</code>	168
<code>\makeClusters</code>	154	<code>\parenthesize</code>	206, 668
<code>\marcato</code>	109	<code>\partcombine</code>	163
<code>\mark</code>	100, 216	<code>\partcombineApart</code>	164
<code>\markalphabet</code>	681	<code>\partcombineAutomatic</code>	164
<code>\markletter</code>	681	<code>\partcombineChords</code>	164
<code>\markup</code>	216, 219, 220, 221	<code>\partcombineSoloI</code>	164
<code>\markuplist</code>	220, 233	<code>\partcombineSoloII</code>	164
<code>\maxima</code>	40, 51	<code>\partcombineUnisono</code>	164
<code>\medium</code>	645	<code>\partial</code>	66, 134, 136
<code>\melisma</code>	247	<code>\path</code>	669
<code>\melismaEnd</code>	247	<code>\pattern</code>	682
<code>\mergeDifferentlyDottedOff</code>	158	<code>\pes</code>	430
<code>\mergeDifferentlyDottedOn</code>	158	<code>\phrasingSlurDashed</code>	122
<code>\mergeDifferentlyHeadedOff</code>	158	<code>\phrasingSlurDashPattern</code>	122
<code>\mergeDifferentlyHeadedOn</code>	158	<code>\phrasingSlurDotted</code>	122
<code>\mf</code>	112	<code>\phrasingSlurDown</code>	122
<code>\midi</code>	449	<code>\phrasingSlurHalfDashed</code>	122
<code>\minor</code>	20	<code>\phrasingSlurHalfSolid</code>	122
<code>\mixolydian</code>	20	<code>\phrasingSlurNeutral</code>	122

<code>\phrasingSlurSolid</code>	122	<code>\sfz</code>	112
<code>\phrasingSlurUp</code>	122	<code>\sharp</code>	674
<code>\phrygian</code>	20	<code>\shiftOff</code>	158
<code>\pitchedTrill</code>	133	<code>\shiftOn</code>	158
<code>\portato</code>	109	<code>\shiftOnn</code>	158
<code>\postscript</code>	230, 669	<code>\shiftOnnn</code>	158
<code>\powerChords</code>	360	<code>\shortfermata</code>	109
<code>\pp</code>	112	<code>\showKeySignature</code>	375
<code>\ppp</code>	112	<code>\showStaffSwitch</code>	305
<code>\pppp</code>	112	<code>\signumcongruentiae</code>	109
<code>\ppppp</code>	112	<code>\simple</code>	647
<code>\prall</code>	109	<code>\skip</code>	53
<code>\pralldown</code>	109	<code>\slashed-digit</code>	683
<code>\prallmordent</code>	109	<code>\slashedGrace</code>	102
<code>\prallprall</code>	109	<code>\slurDashed</code>	119
<code>\prallup</code>	109	<code>\slurDashPattern</code>	120
<code>\predefinedFretboardsOff</code>	355	<code>\slurDotted</code>	119
<code>\predefinedFretboardsOn</code>	355	<code>\slurDown</code>	119
<code>\property-recursive</code>	682	<code>\slurHalfDashed</code>	120
<code>\pt</code>	580	<code>\slurHalfSolid</code>	120
<code>\pushToTag</code>	477	<code>\slurNeutral</code>	119
<code>\put-adjacent</code>	661	<code>\slurSolid</code>	119
<code>\quilisma</code>	423, 430	<code>\slurUp</code>	120
<code>\quoteDuring</code>	192, 195	<code>\small</code>	200, 225, 647
<code>\raise</code>	226, 661	<code>\smallCaps</code>	648
<code>\relative</code>	2, 12, 304	<code>\smaller</code>	223, 225, 648
<code>\RemoveEmptyStaves</code>	185	<code>\snappizzicato</code>	109
<code>\RemoveEmptyStaves</code>	186	<code>\sostenutoOff</code>	307
<code>\removeWithTag</code>	477	<code>\sostenutoOn</code>	307
<code>\repeat</code>	134	<code>\southernHarmonyHeads</code>	36
<code>\repeat percent</code>	145	<code>\southernHarmonyHeadsMinor</code>	36
<code>\repeat tremolo</code>	147	<code>\sp</code>	112
<code>\repeat unfold</code>	143	<code>\spp</code>	112
<code>\repeatTie</code>	48, 137, 264	<code>\staccatissimo</code>	109
<code>\replace</code>	646	<code>\staccato</code>	109
<code>\rest</code>	51	<code>\startGroup</code>	211
<code>\retrograde</code>	13	<code>\startStaff</code>	178, 181
<code>\reverseturn</code>	109	<code>\startTrillSpan</code>	132
<code>\revert</code>	572	<code>\stemDown</code>	207
<code>\revertTimeSignatureSettings</code>	61	<code>\stemNeutral</code>	207
<code>\rfz</code>	112	<code>\stemUp</code>	207
<code>\rheel</code>	109	<code>\stencil</code>	683
<code>\right-align</code>	225, 661	<code>\stopGroup</code>	211
<code>\right-brace</code>	683	<code>\stopped</code>	109
<code>\right-column</code>	661	<code>\stopStaff</code>	178, 181, 185
<code>\rightHandFinger</code>	357	<code>\stopTrillSpan</code>	132
<code>\roman</code>	647	<code>\storePredefinedDiagram</code>	348
<code>\rotate</code>	662	<code>\stringTuning</code>	332
<code>\rounded-box</code>	228, 670	<code>\strophae</code>	423, 430
<code>\rtoe</code>	109	<code>\strut</code>	683
<code>\sacredHarpHeads</code>	36	<code>\sub</code>	224, 648
<code>\sacredHarpHeadsMinor</code>	36	<code>\super</code>	224, 648
<code>\sans</code>	647	<code>\sustainOff</code>	307
<code>\scale</code>	670	<code>\sustainOn</code>	307
<code>\scaleDurations</code>	47, 69	<code>\tabChordRepeats</code>	321
<code>\score</code>	445, 449, 673	<code>\tabFullNotation</code>	320
<code>\segno</code>	109	<code>\table-of-contents</code>	473, 685
<code>\semiflat</code>	674	<code>\tag</code>	477
<code>\semiGermanChords</code>	396	<code>\taor</code>	375
<code>\semisharp</code>	674	<code>\teeny</code>	200, 225, 649
<code>\sesquiflat</code>	674	<code>\tempo</code>	63
<code>\sesquisharp</code>	674	<code>\tenuto</code>	109
<code>\set</code>	78, 569	<code>\text</code>	649
<code>\sf</code>	112	<code>\textLengthOff</code>	56, 213
<code>\sff</code>	112	<code>\textLengthOn</code>	56, 213

<code>\textSpannerDown</code>	214
<code>\textSpannerNeutral</code>	214
<code>\textSpannerUp</code>	214
<code>\thumb</code>	109, 202
<code>\tied-lyric</code>	675
<code>\tieDashed</code>	48
<code>\tieDotted</code>	48
<code>\tieDown</code>	48
<code>\tieNeutral</code>	48
<code>\tieSolid</code>	48
<code>\tieUp</code>	48
<code>\time</code>	58, 78
<code>\times</code>	42, 69
<code>\tiny</code>	200, 225, 649
<code>\tocItem</code>	473
<code>\translate</code>	226, 662
<code>\translate-scaled</code>	226, 662
<code>\transparent</code>	683
<code>\transpose</code>	10, 12
<code>\transposedCueDuring</code>	198
<code>\transposition</code>	22, 192
<code>\treCorde</code>	307
<code>\triangle</code>	230, 671
<code>\trill</code>	109, 132
<code>\tupletDown</code>	42
<code>\tupletNeutral</code>	42
<code>\tupletUp</code>	42
<code>\turn</code>	109
<code>\tweak</code>	573
<code>\type</code>	562
<code>\typewriter</code>	649
<code>\unaCorda</code>	307
<code>\underline</code>	223, 650
<code>\unfoldRepeats</code>	489
<code>\unHideNotes</code>	204
<code>\unset</code>	570
<code>\upbow</code>	109
<code>\upbow</code>	314
<code>\upmordent</code>	109
<code>\upprall</code>	109
<code>\upright</code>	650
<code>\varcoda</code>	109
<code>\vcenter</code>	663
<code>\verbatim-file</code>	683
<code>\verylongfermata</code>	109
<code>\virga</code>	423, 430
<code>\virgula</code>	422
<code>\voiceFourStyle</code>	158
<code>\voiceNeutralStyle</code>	158
<code>\voiceOne</code>	155
<code>\voiceOne ... \voiceFour</code>	155
<code>\voiceOneStyle</code>	158
<code>\voiceThreeStyle</code>	158
<code>\voiceTwoStyle</code>	158
<code>\void</code>	495
<code>\vspace</code>	663
<code>\walkerHeads</code>	36
<code>\walkerHeadsMinor</code>	36
<code>\whiteout</code>	684
<code>\with</code>	555, 560
<code>\with-color</code>	205, 684
<code>\with-dimensions</code>	684
<code>\with-link</code>	684
<code>\with-url</code>	671
<code>\woodwind-diagram</code>	677

<code>\wordwrap</code>	227, 664
<code>\wordwrap-field</code>	663
<code>\wordwrap-internal</code>	685
<code>\wordwrap-lines</code>	233, 685
<code>\wordwrap-string</code>	664
<code>\wordwrap-string-internal</code>	685

Affichage d'expressions musicales

Affichage d'expressions musicales..... 495, 575

Application d'une fonction a tous les obj

Application d'une fonction à tous les objets de
mise en forme..... 552

Construction d'un markup en Scheme

Construction d'un markup en Scheme..... 118, 119,
281

Definition d'une nouvelle commande de l

Définition d'une nouvelle commande de liste de
markups..... 234

Espacement

Espacement 534

Fonctions de rappel

Fonctions de rappel 593

Fonctions musicales

Fonctions musicales 602, 603

Interfaces pour programmeurs

Interfaces pour programmeurs..... 593

References bibliographiques

Références bibliographiques..... 393, 398

Tutoriel Scheme

Tutoriel Scheme..... 549

|

| 99

~

~ 47

1

15ma 21

8

8va	21
8ve	21

A

a due	163
à l'italienne, papier	498
absolues, hauteurs	1
absolues, octaves	1
accent	109, 110, 687
accepts	562
acciaccatura	722
acciaccature	102
acciaccature, multi-note	106
accidentalStyle	722
accidentel, quart de ton	7
accidentelles automatiques	23
accidentelles, altérations	23
accolade verticale	172
accolades, taille	234
accord arpégé	129
accord et mode relatif	150
accord nommés et diagrammes de fret	345
accord vide	150
accord, additions	388
accord, altération d'un degré	389
accord, carrures pour cordes fretées	348
accord, diagrammes	334, 344
accord, diagrammes automatiques	354
accord, inversion	389
accord, personnalisation du chiffrage	393
accord, répétition	151, 321
accord, spécifier la basse	389
accord, suppression d'un degré	389
accordage non occidental	437
accordage personnalisé	332
accordages de banjo	362
accordages prédéfinis et cordes fretées	331
accordéon, symboles de registre	308
accordéon, tirettes	308
accords	149, 391
accords chiffrés, exceptions	396
accords et altérations	29
accords et couleur	206
accords et liaisons de tenue	48
accords et octave relative	4
accords jazz	393
accords nommés en MIDI	488
accords, chiffrage	385
accords, doigtés	202
accords, mode	385
accords, modificateur	386
accords, No Chord	391
accords, noms	385
accords, répartition sur plusieurs portées avec \autochange	305
accords, suppression des répétitions	392
addChordShape	348
addChordShape	722
adding a white background to text	684
addInstrumentDefinition	191, 198, 722
additionalPitchPrefix	394
additions à un accord	388

addQuote	192, 722
aeolian	20
affinage (tweak)	573
afterGrace	102, 722
afterGraceFraction	692
Aiken, tête de note	36
aikenHeads	36
aikenHeadsMinor	36
ajout de texte	213
ajustement (tweak)	573
ajustement des symboles de portée	581
al niente	115
alias	562
alignAboveContext	564
alignBelowContext	261, 564
aligné, ragged	496
alignement des markups	225
alignement du numéro de mesure	98
alignement du texte	225
alignement du texte, commandes	228
alignement et cadence	107
alignement horizontal du texte	225
alignement sur un objet	596
alignement vertical du texte	226
alist	690
allowPageTurn	722
altérables, objets et propriétés	692
altération	5, 417
altération de précaution	5
altération de précaution style <i>modern voice</i>	26
altération de précaution, style <i>modern</i>	26
altération entre parenthèses	5
altération et liaison de tenue	6
altération et musica ficta	417
altération, style	23
altération, style <i>dodecaphonic</i>	28
altération, style <i>modern</i>	26
altération, style <i>modern cautionary</i>	26
altération, style <i>no reset</i>	29
altération, style par défaut	23
altération, trilles avec hauteur explicite	133
altérations	421
altérations et accords	29
altérations et cadence	67
altérations et notes simultanées	29
altérations multivoix	26, 28
altérations style <i>piano</i>	27
altérations style <i>piano cautionary</i>	27
altérations, style <i>default</i>	25
altérations, style <i>forget</i>	29
altérations, style <i>modern</i>	25
altérations, style <i>modern-cautionary</i>	25
altérations, style <i>neo-modern</i>	27
altérations, style <i>teaching</i>	28
altérations, style <i>voice</i>	25
alterBroken	722
alternative et liaison de prolongation	137
alternative et paroles	262
alternative et texte	142
alternative, bascule sur une mélodie	270
alternative, reprise	134
Amazing Grace : exemple pour cornemuse	375
ambitus	30
amplitude	30
anacrouse	66

anacrouse et reprise	136
analyse lexicale	691
analyse musicologique	211
analyseur syntaxique	692
ancien, crochet	415
ancienne, métrique	413
anciennes altérations	417
anciennes ligatures	418
anciennes têtes de note	414
anciennes, clefs	412
anciens silences	416
annotate-spacing	546
annotation	221
annulation d'un override	572
appendToTag	722
applyContext	722
applyMusic	722
applyOutput	722
appoggiatura	723
appoggiature	102
arabe, noms des notes en	439
armure	5, 19, 417, 421
armure, visibilité après changement explicite	589
armures arabes	439
arpège	129
arpège, style parenthèse	131
arpège, symboles spéciaux	129
arpeggio	129
arpeggioArrowDown	129
arpeggioArrowUp	129
arpeggioBracket	129
arpeggioNormal	129
arpeggioParenthesis	129
arpeggioParenthesisDashed	129
arrow-head	230
articulation, liaison	119
articulation-event	194
articulations	109
articulations et MIDI	488
artificiels, harmoniques	315
aspect d'un symbole de demi-bémol	439
assertBeamQuant	723
assertBeamSlope	723
associatedVoice	240
associatedVoice	242, 270
aug	387
auto-first-page-number	505
autoBeaming	78
autoBeamOff	75
autoBeamOn	75
autochange	303, 723
autochange et mode relatif	304
automaticBars	590
Automatique, changement de portée	303
automatique, diagramme d'accord	354
automatique, diagramme de fret	354
automatiques, altérations accidentelles	23

B

backslashed digits.....	679
bagpipe.....	375
balise.....	477
Balloon_engraver	207

balloonGrobText	207, 723
balloonLengthOff	207
balloonLengthOn	207
balloonText	207, 723
banjo, accordages	362
banjo, tablature	331
banjo, tablature pour	317
banjo, tablatures	362
banjo-c-tuning	362
banjo-modal-tuning	362
banjo-open-d-tuning	362
banjo-open-dm-tuning	362
bar	90, 94, 723
barCheckSynchronize	99
BarNumber	95
barNumberCheck	99, 723
barNumberVisibility	95
baroque, ornamentation	109, 687
barre de mesure invisible	90
barre de mesure manuelle	90
barre de mesure par défaut, modification du style	94
barre finale	90
barré, indication de	335
barré, indication de la position	358
barres de mesure	90
barres de mesure et cadence	67
barres de mesure, suppression	67, 590
barres de mesure, symboles au dessus de	216
barres de reprise	90
Bartók pizzicato	316
bartype	94
bas de page, notes	463
base-shortest-duration	534
baseMoment	78
basse chiffrée	399
basse chiffrée, alignement	404
basse chiffrée, lignes d'extension	402
basse continue	399
basse d'un accord	389
battements par minute	63
batterie	363, 365
batterie, portée	171
beamExceptions	78
beatStructure	78
bécarre	5
bémol	5
bendAfter	124, 723
Bézier, points de contrôle d'une courbe	599
binding-offset	503
bisbigliando	312
Bison	692
blanc	450
blank-after-score-page-force	504
blank-last-page-force	504
blank-page-force	505
bloc de texte	213
BNF	692
bold	223
bookOutputName	723
bookOutputSuffix	723
bookTitleMarkup	459
bottom-margin	499
bouché	109, 687
box	228

bracket	117, 228, 308
break-align-symbols	596
break-visibility	587
breakable	76
breakbefore	456
breathe	123
breathe	723
breve	40, 51
broderies	102
bulle	207

C

cadence	107
cadence et alignement	107
cadence et altérations	67
cadence et barres de mesure	67
cadence et ligatures	67
cadence et numéros de mesure	67
cadence et saut de ligne	69
cadence et saut de page	69
cadenza	107
cadenzaOff	67
cadenzaOn	67
callback	690
calque	587
calques	156
cantique	287
capo	339
caractères réservés, impression	221
caractères spéciaux	480
caractères spéciaux en mode markup	221
caractères, taille	223
cases	320
center-align	225
center-column	227
centering a column of text	651
centrage des nuances entre deux portées de piano	301
centrage du texte sur la page	227
césure	124
césures	421
change	302
changement d'instrument	191
changement de fonte	223
changement de nom d'instrument	190
Changement de portée automatique	303
changement de portée et collision	302
changement de portée forcé	302
changements de portée	305
changements de portée manuels	302
changing direction of text columns	652
check-consistency	502
chevrons	149
chiffage d'accords, exceptions	396
chiffage d'accords, personnalisation	393
chiffage de mesure	58
chiffrages d'accords	391
chiffre indicateur de mesure	58
chiffres de mesure multiples	556
choeur, citation d'instrument	281
choeur, partition de	275
choeur, système	172
chorale, clef de ténor	17
chord, power	360
chordChanges	392
chordmode	12, 345
chordmode	385
chordNameExceptions	395
chordNameLowercaseMinor	394
ChordNames	345
chordNameSeparator	394
chordNoteNamer	394
chordPrefixSpacer	396
chordRepeats	723
chordRootNamer	394
Christian Harmony, tête de note	36
chutes	124
circle	228
circling text	666
circulus	687
citation	192
citation, fin	198
clavier, portées pour instrument à	301
claviers, centrage des nuances	301
clé d'ut	16
clé de fa	16
clé de sol	16
clef	5
clef	16, 723
clef de tablature	334
clef et transposition	17
clef moderntab	334
clef, percussion	363
clef, visibilité après changement explicite	589
clef, visibilité de l'octaviation	591
clefs	412, 420
clés	412
clés anciennes	16
clés, portées pour instrument à	301
clôture	690
cluster	154
coda	101, 109, 687
coda sur une barre de mesure	216
collision de notes	158
collision et changement de portée	302
collision et numéro de mesure	99
collision sur empilement de notes	153
collisions	158
collisions, ignorer	153, 163
colonnes de texte	227
color	205
coloration d'objets	205
coloration de notes	205
colorier des voix	158
coloring text	684
colorisation d'objets	205
colorisation de notes	205
column	227
combinaison de parties	163
combine	230
comma	443
commandes d'alignement du texte	228
commentaire textuel	221
common-shortest-duration	534
Completion_heads_engraver	72
Completion_rest_engraver	72
complexe, métrique	71
composite, métrique	69
compoundMeter	723

compressFullBarRests	55, 56
concatenating text	652
condenser les silences	58
condition et markup	462
consists	562
contexte, création	551, 562
contexte, définition en MIDI	487
contexte, modification des propriétés par défaut ..	557
contextes, durée de vie	552
contextes, maintien actif	552
contextes, ordonnancement	564
contrôle des hauteurs	9
controlling general text alignment	654
controlpitch	9
copyright	482
corde à vide, indication	314
cordes d'orchestre	314
cordes frettées, carrures d'accord	348
cordes frettées, doigtés main droite	357
cordes frettées, harmoniques	359
cordes frettées, indication de la position et du barré	358
cordes frettées, notes étouffées	359
cordes frettés et accordages prédéfinis	331
cordes numérotées	317
cordes, écriture pour	314
cornemuse	375
cornemuse : exemple	375
couches	156
couleur d'objet	587
couleur rgb	205
couleur rvb	205
couleur, notes d'un accord	206
couleurs	205
couleurs x11	205
couleurs, liste	619
coup d'archet	109
coups de gratte, indication	73
courbes	119
cr	112
creating empty text objects	681
creating horizontal spaces in text	656
creating text fractions	679
creating vertical spaces in text	663, 683
création de contextes	552
cresc	113
crescendo	112
crescendo-event	194
crescHairpin	113
crescTextCresc	113
crochet de nolet, positionnement	42
crochet de regroupement de notes	211
crochet vertical	172
crochets	211, 415
crochets de phrasé	211
crochets, notes entre	206
croix, tête de note	33
cross	33
cross-staff tremolo	148
crossStaff	723
cueClef	195, 723
cueClefUnset	723
cueDuring	195, 724
cueDuringWithClef	195, 724
CueVoice	195

currentBarNumber	94
currentBarNumber	108
custodes	411
custos	411

D

D.S al Fine	101
deadNote	724
décalage de note	158
décalage de voix	158
décoration du texte	228
decr	112
decresc	113
decrescendo	112
default	23, 25
<i>default</i> , style d'altérations	25
defaultBarType	94
defaultNoteHeads	724
defaultTimeSignature	59
défaut, durée	40
définition de sortie	692
délimitation, systèmes imbriqués	176
demi-bémol	8, 439
demi-bémols	5
demi-dièse	8
demi-dièses	5
denies	562
dérégations en mode paroles	239
dessin des symboles de portée	581
diagrammes d'accord pour instrument fretté	334
diagrammes de fret	334
diagrammes de fret et transposition	346
diagrammes de fret personnalisés	341
diagrammes de fret personnalisés, ajout	347
diagrammes personnalisés de fret	334, 341
dièse	5
dim	113, 387
dimHairpin	113
diminuendo	112
dimTextDecr	113
dimTextDecresc	113
dimTextDim	113
displayLilyMusic	724
displayMusic	724
disponibilité des fontes	236
distance absolue	580
distance entre deux portées de piano	305
distance entre les portées	518
distance extensible	580
distance relative	580
divisio	421
division de notes	72
division de silences	72
division et paroles	266
divisiones	421
dodecaphonic	28
<i>dodecaphonic</i> , style d'altérations	28
doigté	201
doigtés des accords	202
doigtés et numéros de corde	317
doigtés et silences multimesure	58
doigtés main droite et cordes frettées	357
doigtés, ajout à des diagrammes de fret	355
dorian	20

dorien	20
<code>dotsDown</code>	41
<code>dotsNeutral</code>	41
<code>dotsUp</code>	41
double barre	90
double bémol	5
double dièse	5
double pause	51
double point	41
doublément pointée, note	41
<code>draw-circle</code>	230
<code>draw-line</code>	230
drawing a line across a page	666
drawing beams within text	665
drawing boxes with rounded corners	667
drawing boxes with rounded corners around text	670
drawing circles within text	666
drawing lines within text	667
drawing paths	669
drawing solid boxes within text	667
drawing triangles within text	671
<code>drummode</code>	171
<code>DrumStaff</code>	171
durée d'un silence	51
durée par défaut	40
durées	40
durées, échelonnement	46
<code>dynamic</code>	117
<code>dynamic-event</code>	194
<code>dynamicDown</code>	114
<code>DynamicLineSpanner</code>	114
<code>dynamicNeutral</code>	114
<code>dynamicUp</code>	114

E

<i>easy play</i> , têtes de note	34
<code>easyHeadsOff</code>	34
<code>easyHeadsOn</code>	34
échelonnement de musique	47
échelonnement des durées	46
écrire la musique en parallèle	168
encadrement du texte	228
encapsulated postscript	484
enclosing text in a box with rounded corners	670
enclosing text within a box	642
<code>endSpanners</code>	724
éolien	20
EPS, format de sortie	484
<code>epsfile</code>	230
espace	450
espacement au sein d'un système	518
espacement autour du texte	229
espacement des paroles	256
espacement entre les portées	518
espacement horizontal	533
espacement horizontal, affinage	600
espacement vertical	518
espacement, affichage des valeurs	546
espaces dans les paroles	246
espaces, dans les paroles	239
<i>espressivo</i>	109
<i>espressivo</i>	113

<i>espressivo</i>	687
étiquette	221
étiquette de texte	213
étiquette et silence multimesure	55
étiquette textuelle	221
événementielle, note de bas de page	465
<code>eventChords</code>	724
exceptions, chiffrage d'accords	396
exemple de musique arabe	442
<code>expandFullBarRests</code>	55, 56
expansion de reprise	143
<code>explicitClefVisibility</code>	589
<code>explicitKeySignatureVisibility</code>	589
exposant	224
expressions markup	221
extenseur	214, 250
extenseur, mise en forme	214

F

<code>f</code>	112
fantômes, notes	206
<code>featherDurations</code>	89, 724
<code>fermataMarkup</code>	55, 56
Feta, fonte	620
<code>ff</code>	112
<code>fff</code>	112
<code>ffff</code>	112
<code>fffff</code>	112
<code>fill-line</code>	227
<code>filled-box</code>	230
fin alternative	134
fin de citation	198
fin de réplique	198
finalis	421
<code>finger</code>	201
<code>first-page-number</code>	505
flageolet	109, 687
Flex	691
fois, première	134
<code>followVoice</code>	305
fondamentale	386
<code>font-interface</code>	201, 234
<code>font-size</code>	200, 201
fonte	690
fonte Feta	620
fonte, changement	223
fonte, définition de la taille	509
fontes disponibles	236
fontes et markup	234
fontes, choix par défaut	236
fontes, familles	224
fontes, généralités	234
<code>fontsize</code>	223
<code>fontSize</code>	200
<code>footnote</code>	724
footnotes	463
<code>forget</code>	29
<i>forget</i> , style d'altérations	29
format de repère	100
formatage du numéro de mesure	96
Forme Semai	441
<code>four-string-banjo</code>	362
<code>fp</code>	112

Frenched scores	185
fret et transposition	346
fret, ajout de diagrammes personnalisés	347
fret, ajout de doigtés aux diagrammes	355
fret, diagrammes	334, 344
fret, diagrammes automatiques	354
fret, diagrammes avec noms d'accord	345
fret, diagrammes personnalisés	334, 341
fret-diagram	336
fret-diagram et markup	336
fret-diagram-interface	341
fret-diagram-terse	338
fret-diagram-terse markup	338
fret-diagram-verbose	339
fret-diagram-verbose markup	339
FretBoards	344
frets	320
Funk, tête de note	36
funkHeads	36
funkHeadsMinor	36
fusion de notes	158

G

general-align	226
glissando	125
glissando, indication en tablature	324
globale, variable	692
glyphe	690
glyphes musicales	101
gouttière	503
grace	724
grammaire de LilyPond	692
graphique, objet	691
graphiques, intégration	230
graphiques, tracé d'objets	228
graphisme dans la notation	230
graphisme, tracé	228
gras	223
graveur, affectation à un contexte	562
GregorianTranscriptionStaff	171
grégorien, articulations	422
grégorien, ligatures de neumes carrés	423
Grid_line_span_engraver	208
Grid_point_engraver	208
gridInterval	208
grob	691
grob, propriétés	571
grob-interface	691
grobdescriptions	724
grobs, superposition	587
grobs, visibilité	586
groupement de mesures	83
groupement de pulsations	83
groupements de note manuels	86
grow-direction	89
grupetto	109, 687
guidon	411
guillemets dans les paroles	246
guillemets en mode markup	221
guillemets, dans les paroles	239
guitare basse, tablature	331
guitare, coups de gratte	73
guitare, grille d'accords	73
guitare, tablature pour	317

guitare, tête de note	33
-----------------------------	----

H

Hal Leonard	34
halign	225
hampe barrée	104
hampe descendante	207
hampe montante	207
hampe neutre	207
hampe, enjambement portées	305
hampe, orientation	207
hampes	207
hampes invisibles	207
Harmonica Sacra, tête de note	36
harmonicByFret	724
harmonicByRatio	724
harmonicNote	725
harmonicsOn	725
harmonique, tête de note	33
harmoniques artificiels	315
harmoniques et cordes frettées	359
harmoniques naturels	315
harmoniques, indication en tablature	324
harpe sacrée, tête de note	36
harpe, diagrammes de pédales	312
harpe, pédales	312
harpes	312
hauteur, nom par défaut	5
hauteurs	1
hauteurs en MIDI	488
hbracket	228
hideKeySignature	375
hideNotes	204
hideStaffSwitch	305
horizontal, espacement	533
horizontal-shift	504
Horizontal_bracket_engraver	211
horizontally centering text	650
hufnagel	408, 409
huge	200, 225
hymne	287

I

identificateurs	450
images, intégration	230
imbrication de reprises	140
imbrication de systèmes	176
importing stencils into text	683
impression de caractères réservés	221
impression de caractères spéciaux	221
impression, ordre	587
improvisation	39
improvisationOff	39, 73
improvisationOn	39, 73
inaltérables, objets	691
inaltérables, propriétés	691
include-settings	480
inclusion de fichiers	473
incomplète, mesure	66
indent	189, 504, 537
indépendant, texte	219
indication d'octave relative	2

indication de corde à vide	314
indication de nuance, personnalisation	117
indication de tempo et nolet	46
indication du barré	335
indication textuelle	216
indications d'archet	314
indice	224
info-bulle	207
inlining an Encapsulated PostScript image	667
inner-margin	503
inserting music into text	673
inserting PostScript directly into text	669
inserting URL links into text	671
inStaffSegno	725
instrument, changement	191
instrument, changement de nom	190
instrument, citation	281
instrument, nom d'	486
instrument, noms	188
instrument, noms abrégés	188
instruments à vent	372
instruments transpositeurs	10
instrumentSwitch	191, 725
intégration d'objet graphique	228
intégration de graphiques	230
inter-portée, ligatures	302
inter-portée, notes	302
interface	691
interfaces de rendu	566
intervalles de comma	443
intervalles medium	438
inversion	13
inversion	725
invisible, barre de mesure	90
invisibles, notes	204
ionian	20
ionien	20
italic	223
italique	223

J

jazz, accords	393
justifié, texte	227
justified-lines	233
justify	227
justifying lines of text	685
justifying text	657

K

keepWithTag	725
key	19, 36, 725
Kievan	430
KievanStaffContext	430
KievanVoiceContext	430
kiévien	430
kiéviennne, clef	431
kiéviennnes, altérations	432
kiéviennnes, têtes de note	431
killCues	198, 725
kirchenpausen	56

L

label	725
laissez vibrer	48
laissezVibrer	48
landscape, papier	498
language	725
languageRestore	725
languageSaveAndChange	725
langues, noms de note	7
large	200, 225
larger	223, 225
last-bottom-spacing	501
layers	587
layout file	509
left aligning text	658
left-align	225
left-margin	502
legato	119
levée	66
levée dans une reprise	136
lexer	691
liaison d'articulation	119
liaison de prolongation	47
liaison de prolongation et alternative	137
liaison de prolongation et reprise	137
liaison de prolongation et reprise avec alternative	48
liaison de prolongation, apparence	48
liaison de prolongation, pointillés	48
liaison de prolongation, tirets	48
liaison de tenue et accords	48
liaison de tenue et altération	6
liaison de tenue et répétition	48
liaison et reprise	140
liaison, au-dessous des notes	119
liaison, au-dessus des notes	119
liaison, densité des tirets	120, 122
liaison, laissez vibrer	48
liaison, modification	599
liaison, orientation manuelle	119
liaison, style de trait	119, 122
liaison, tirets	119, 122
liaison, tirets et trait continu	120, 122
liaison, trait plein	119, 122
liaison, trait pointillé	119, 122
liaisons d'articulation multiples	119
liaisons d'articulation simultanées	119
liaisons de phrasé	122
liaisons de phrasé multiples	122
liaisons de phrasé simultanées	122
liaisons de prolongation, positionnement	48
liaisons, dans les paroles	246
ligature	410
ligature coudée	76
ligature en fin de partition	84
ligature en fin de voix polyphonique	84
ligature en soufflet	89
ligature et cadence	67
ligature et mélisme	76
ligature et métrique	59
ligature et musique polymétrique	69
ligature et saut de ligne	76
ligature inter-portée	302
ligature manuelle et notes d'ornement	86

ligature manuelle, orientation.....	86
ligature, \partcombine et \autoBeamOff.....	77
ligatures anciennes.....	418
ligatures blanches.....	418
ligatures et neumes.....	423
ligatures in text.....	652
ligatures manuelles.....	75, 86
ligatures te paroles.....	78
ligatures, définition de règles.....	75
ligatures, personnalisation.....	75
ligatures, subdivision.....	82
ligne supplémentaire.....	178
ligne, longueur.....	537
ligne, passer à la suivante.....	510
lignes.....	125
lignes de portée, épaisseur.....	178
lignes de portée, nombre de.....	178
line-width	502, 537
liste associative.....	690
liste des couleurs.....	619
liturgie et musique.....	287
locrian	20
locrien.....	20
longa	40, 51
longueur de note.....	40
losange, tête de note.....	33, 315
lower	226
lowering text.....	659
ly:minimal-breaking	514
ly:one-line-breaking	514
ly:optimal-breaking	513
ly:page-turn-breaking	513
lydian	20
lydien.....	20

M

m	387
magnify	223
magnifying text.....	644
magstep	200, 580
main droite, doigtés pour cordes frettées.....	357
maj	387
majeur.....	20
major	20
majorSevenSymbol	394
makam	443
makamlar	443
make-dynamic-script	118
make-pango-font-tree	236
makeClusters	154, 725
makeDefaultStringTuning	725
mandoline, tablature.....	331
manuelle, barre de mesure.....	90
Manuels	1
maqam.....	438, 443
maqams.....	438
marcato.....	109, 110, 687
marges, texte qui dépasse.....	213
mark	100, 216, 725
markup	216, 219, 220, 221
markup conditionnel.....	462
markup et fontes.....	234
markup et fret-diagram.....	336

markup multi-page.....	233
markup multiligne.....	227
markup, centrage sur la page.....	227
markup, commandes d'alignement du texte.....	228
markup, encadrement du texte.....	228
markup, expressions.....	221
markup, inclusion de musique.....	231
markup, inclusion de partition.....	233
markup, ornementation du texte.....	228
markup, rembourrage du texte.....	229
markup, syntaxe.....	221
markup, texte au kilomètre.....	227
markup, texte justifié.....	227
markup, texte multi-page.....	233
markup-markup-spacing	501
markup-system-spacing	501
markuplist	220, 233
markups, alignement.....	225
masquage de portée.....	185
masquage des hampes.....	207
masquées, notes.....	204
max-systems-per-page	504
maxima	40, 51
measureLength	78
measureLength	108
measurePosition	66
measurePosition	108
Medicaea, Editio.....	408, 409
mélisme.....	247, 250
mélisme et ligature.....	76
mélodie alternative.....	270
mélodie d'une portée à une autre.....	305
mélodie, affichage du rythme seul.....	73
mensural.....	408, 409
MensuralStaff	171
MensuralStaffContext	411, 417
MensuralVoiceContext	411, 417
mergeDifferentlyDottedOff	158
mergeDifferentlyDottedOn	158
mergeDifferentlyHeadedOff	158
mergeDifferentlyHeadedOn	158
merging text.....	651
mesure de silence.....	51
mesure entière de silence.....	54
mesure incomplète.....	66
mesure, numéro de.....	94, 108
mesure, numérotation et reprises.....	97, 139
mesure, répétition de.....	145
mesure, subdivision.....	83
mesures à compter.....	54
mesures, vérification des limites.....	99
métrique.....	58
métrique composite.....	69, 71
métrique double.....	69
métrique et ligature.....	59
métrique par défaut.....	59
métrique polymétrique.....	69
métrique, retour au propriétés par défaut.....	61
métrique, styles.....	59
métrique, visibilité.....	58
métriques arabes.....	441
métronomie, indication.....	63
mf	112
microtonalité en MIDI.....	488
microtonalités.....	8

MIDI	485
MIDI et articulations	488
MIDI et définition de contexte	487
MIDI et reprises	489
MIDI et transposition	22
MIDI et trilles	488
MIDI, accords nommés	488
MIDI, hauteurs	488
MIDI, microtonalité	488
MIDI, quart de ton	488
MIDI, Rythme	488
min-systems-per-page	504
mineur	20
minimumFret	320
minimumPageTurnLength	513
minimumRepeatLengthForPageTurn	514
minor	20
minorChordModifier	395
mirroring markup	670
mixed	308
mixolydian	20
mixolydien	20
modale, inversion	15
modale, transposition	14
modales, transformations	14
modalInversion	15, 725
modalTranspose	14, 725
mode	692
mode markup et caractères spéciaux	221
mode markup et guillemets	221
mode relatif et accord	150
mode relatif et autochange	304
modèle de musique arabe	442
modern	26
<i>modern cautionary</i> , style d'altération	26
<i>modern voice</i> , style d'altération de précaution	26
<i>modern</i> , style d'altération	25, 26
<i>modern</i> , style d'altération de précaution	26
modern-cautionary	26
<i>modern-cautionary</i> , style d'altération	25
modern-voice	26
modern-voice-cautionary	26
moderntab, clef	334
modes	20
modes anciens	20
modification du style par défaut des barres de mesure	94
modifier des propriétés	569
mordant	109, 687
motet	275
mouvements, plusieurs	446
mp	112
multi-page, markup	233
multiligne, markup	227
multiligne, texte	227
MultiMeasureRestText	55
multiples voix	158
multiples, liaisons d'articulation	119
multiples, liaisons de phrasé	122
multivoix, altérations	26, 28
musica ficta	417
musicale, citation	281
musicglyph	101
musicMap	725
musicQuotes	692

musique ancienne, masquage de portée	186
musique ancienne, transcription	175
Musique arabe	438
musique arabe, exemple	442
musique dans un markup	231
musique échelonnée	47
musique en parallèle	168
musique entremêlée	168
musique et note de bas de page	464
musique non mesurée	67, 108
musique non mesurée et saut de ligne	69
musique non mesurée et saut de page	69
musique ottomane	443
musique répétitive	143
musique turque	443

N

N.C., symbole	391
name	562
naturel	5
naturels, harmoniques	315
neo-modern	27
<i>neo-modern</i> , style d'altérations	27
neo-modern-cautionary	27
<i>neo-modern-cautionary</i> , style d'altérations	27
neo-modern-voice	28
<i>neo-modern-voice</i> , style d'altération	28
neo-modern-voice-cautionary	28
neomensural	409
neumes carrés et ligatures	423
niente, al	115
no chord, symbole	391
<i>no reset</i> , style d'altérations	29
no-reset	29
noBeam	87
nolet, modification du chiffre	43
nolets	42
nolets, formatage	43
nom d'instrument	188
nom de personnage	268
nom du chanteur	268
nombre de portées	181
noms d'instrument, autres contextes	190
noms d'instrument, centrés	189
noms de note	1
noms de note, hollandais	5
noms de note, langues	7
noms de note, par défaut	5
noms des notes en arabe	439
non mesurée, musique	108
non musicaux, symboles	230
non-ASCII, caractères	480
noPageBreak	725
noPageTurn	725
normal-size-super	224
normalsize	200, 225
notation dans un markup	231
notation et graphisme	230
notation facile	34
notation, explication	207
note d'ornement, synchronisation	106
note invisible	204
note masquée	204

note silencieuse	51
note, décalage	158
note, têtes anciennes	414
note-event	194
Note_heads_engraver	72
notes colorisées dans un accord	206
notes d'ornement	102
notes d'ornement en fin de note	102
notes d'ornement et crochet de nolet	46
notes d'ornement et retouches	103
notes d'ornement, mise en forme	103
notes en couleur	205
notes étouffées et cordes frettées	359
notes fantômes	206
notes inter-portée	302
notes penchées	39
notes pointées	41
notes profilées	36
notes simultanées et altérations	29
notes within text by log and dot-count	672
notes within text by string	672
notes, division	72
notes, longueur	40
notes, noms selon la langue	7
notes, taille	200
notes, taille standard	201
nouveau contexte	551
nouvelle portée	171
nuances	112
nuances éditoriales	117
nuances entre les portées d'un système pianistique	301
nuances entre parenthèses	117
nuances personnalisées	117
nuances successives sur une note tenue	113
nuances suggérées	117
nuances, positionnement vertical	114
null	226
numericTimeSignature	59
numéro de corde	317
numéro de couplet	267
numéro de mesure	108
numéro de mesure à intervalle régulier	95
numéro de mesure et cadence	67
numéro de mesure et collision	99
numéro de mesure et reprise	97, 139
numéro de mesure et reprises	140
numéro de mesure, alignement	98
numéro de mesure, formatage	96
numéros de corde et doigtés	317
numéros de mesure	94
numérotation des mesures, suppression	67

O

objet de propriété	693
objet graphique, interface	691
objet graphique, propriétés	571
objet Scheme	693
objets altérables	692
objets de rendu	691
objets en couleur	205
objets graphiques	566
objets graphiques, tracé	228

objets, couleur	587
objets, surimpression	587
objets, visibilité	586
octave et clef	17
octave relative et accords	4
octave relative, indication	2
octave, spécification	1
octave, vérification	9
octaveCheck	9, 726
octaves absolues	1
octaviation	21
octaviation, visibilité de la clef	591
on-the-fly (à la volée)	462
once	726
oneVoice	155
oratorio	275
orchestre, cordes	314
orgue, marque de pédale	109, 687
ornementation baroque	109, 687
ornementation, symboles	109
ornements	102
ornements et paroles	269
ossia	181, 186
ottava	21, 726
outer-margin	503
output-count	692
output-def	692
output-suffix	692
outside-staff-horizontal-padding	532
outside-staff-padding	532
outside-staff-priority	532
ouvert	109, 687
override ponctuel	573
override, annulation des effets	572
overrideProperty	726
overrideTimeSignatureSettings	726
overriding properties within text markup	681

P

p	112
pad-around	229
pad-markup	229
pad-to-box	229
pad-x	229
padding	568
padding text	660
padding text horizontally	660
page, format	497
page, mise en forme	538
page-breaking	505
page-breaking-system-system-spacing	505
page-count	505
page-spacing-weight	506
pageBreak	726
pages, saut	537
pageTurn	726
palmMute	726
palmMuteOn	726
Pango	234
paper-height	499
paper-width	502
papier, orientation	498
papier, taille	497

<code>parallelMusic</code>	168, 726	phrasé, liaisons de.....	122
parenthèses, altération entre.....	5	phrasé, pour des paroles.....	247
parenthèses, notes entre.....	206	<code>phrasingSlurDashed</code>	122
<code>parenthesize</code>	206, 726	<code>phrasingSlurDashPattern</code>	122, 727
parlato.....	285	<code>phrasingSlurDotted</code>	122
parlato, tête de note.....	33	<code>phrasingSlurDown</code>	122
paroles.....	239	<code>phrasingSlurHalfDashed</code>	122
paroles assignées à une voix.....	155	<code>phrasingSlurHalfSolid</code>	122
paroles divisées (reprises).....	266	<code>phrasingSlurNeutral</code>	122
paroles et liaison de prolongation.....	264	<code>phrasingSlurSolid</code>	122
paroles et ligatures.....	78	<code>phrasingSlurUp</code>	122
paroles et mélodies.....	242	<code>phrygian</code>	20
paroles et ornements.....	269	phrygien.....	20
paroles et répétition.....	258	<code>piano</code>	27
paroles, alignement sur la mélodie.....	240	<i>piano cautionary</i> , style d'altérations.....	27
paroles, alignement sur une mélodie épisodique..	553	piano et altérations.....	27
paroles, blanc.....	53	piano et pédales.....	307
paroles, gestion de l'espace.....	256	piano, nuances entre les portées.....	301
paroles, identificateurs.....	250	<i>piano</i> , style d'altérations.....	27
paroles, mise en forme.....	239	piano, système.....	172
paroles, positionnement.....	252	piano, système pour.....	301
paroles, reprise avec alternative.....	262	<code>piano-cautionary</code>	27
paroles, saut de notes.....	53	<code>PianoStaff</code>	301, 303
paroles, variables.....	250	<code>pipeSymbol</code>	99
parser.....	692	<code>Pitch_squash_engraver</code>	73
parser, variable.....	692	<code>pitchedTrill</code>	133, 727
<code>parseStringResult</code>	692	pitchnames.....	692
<code>partcombine</code>	163, 726	pizzicato, Bartók.....	316
<code>partcombineApart</code>	164	pizzicato, snap.....	316
<code>partcombineAutomatic</code>	164	placeholder events.....	150
<code>partcombineChords</code>	164	placing horizontal brackets around text.....	668
<code>partcombineDown</code>	726	placing parentheses around text.....	668
<code>partcombineForce</code>	726	placing vertical brackets around text.....	666
<code>partCombineListener</code>	692	plein, trait de liaison.....	119, 122
<code>partcombineSoloI</code>	164	plusieurs lignes de texte.....	227
<code>partcombineSoloII</code>	164	plusieurs mouvements.....	446
<code>partcombineUnisono</code>	164	plusieurs pages de texte.....	233
<code>partcombineUp</code>	727	point.....	41
<code>partial</code>	66, 727	point d'arrêt.....	109, 123, 687
partie vocale.....	275	point d'orgue.....	101, 109, 687
parties, combiner des.....	163	point d'orgue et silence multimesure.....	55
partition incluse dans un markup.....	233	point d'orgue sur une barre de mesure.....	216
paths, drawing.....	669	<code>pointAndClickOff</code>	727
paysage, papier.....	498	<code>pointAndClickOn</code>	727
pédale sostenuto.....	307	<code>pointAndClickTypes</code>	727
pédale sustain.....	307	pointe.....	687
pédale sustain, style.....	308	pointée, note.....	41
pédale, indication combinée de.....	308	pointillé, trait de liaison.....	119, 122
pédale, indication graphique de.....	308	points de contrôle et tweak.....	575
pédale, indication textuelle de.....	308	points de contrôle, courbe de Bézier.....	599
pédale, styles d'indications de.....	308	polices, choix par défaut.....	236
pédales de harpe.....	312	polymétrie.....	69
pédales de piano.....	307	polymétrie et ligature.....	69
pédales, diagrammes pour harpe.....	312	polymétrique, partition.....	556
<code>pedalSustainStyle</code>	308	polyphonie.....	158
<code>percent</code>	145	polyphonie, portée simple.....	155
percussion, clef.....	363	ponctuation.....	239
percussion, portée.....	171	ponctuation et paroles.....	239
percussions.....	363, 365	portato.....	109, 110, 687
percussions, nom des notes.....	688	portée à la française.....	181
personnage, indication.....	279	portée à quatre mesures.....	511
petite note.....	102	portée de batterie.....	171
petites notes.....	192, 375	portée de percussion.....	171
petites notes, formater des.....	195	portée de tablature.....	171
Petrucchi.....	408, 409	portée multiple.....	172

portée rythmique	171
portée simple	171
portée vide	185
portée, définition de la taille	509
portée, initialisation	171
portée, instanciation	171
portée, lignes de	178
portée, nouvelle	171
portée, reprise	178
portée, suspension	178
portée, transcription de grégorien	171
portées pour instrument à clavier	301
portées pour instrument à clés	301
portées pour piano	301
portées rythmiques, masquage	186
portées, espacement	518
portées, groupe de	172
portées, nombre variable de	181
portées, regroupement	172
portées, regroupement et imbrication	176
positionnement des paroles	252
positionnement des silences multimesure	56
postscript	230
pouce	109, 687
pouce, doigté	202
pourcent, répétition	145
poussé, indication d'archet	314
pousser l'archet	687
power chords	360
powerChords	360
pp	112
ppp	112
pppp	112
ppppp	112
précaution, altération de	5
predefinedFretboardsOff	355
predefinedFretboardsOn	355
première fois	134
présentation mensurale	175
print-all-headers	506
print-first-page-number	505
print-page-number	505
prob	693
prolongateur	250
prolongation de texte	214
propriété objet	693
propriétés	569
propriétés altérables	692
propriétés communes	691
propriétés d'objet graphique	571
propriétés d'un grob	571
psalmodie	287
pulsation, regroupement	83
pure containers, Scheme	600
pushToTag	727
putting space around text	660

Q

quadrillage temporel	208
qualité d'accord	386
quart de ton	7
quart de ton en MIDI	488
quarts de ton	5

quotedCueEventTypes	194
quotedEventTypes	194
quoteDuring	192, 195, 727

R

r	51
R	54
ragged, aligné	496
ragged-bottom	499
ragged-last	503, 537
ragged-last-bottom	499
ragged-right	502, 537
raise	226
raising text	661
rappel	690
Ratisbona, Editio	409
Référence des propriétés internes	549
referencing page labels in text	684
referencing page numbers in text	682
registres, symboles pour accordéon	308
réglages par défaut, modification	557
regroupement automatique de parties	163
relatif	2
relative	2, 12, 304, 727
religieuse, musique	287
reliure	503
RemoveEmptyStaves	729
removeWithTag	727
renaissance, musique	175
rendu, interfaces de	566
rendu, objets de	691
repeatCommands	140
repeatTie	48
repère manuel	100
repère, format	100
repère, indication de	100
repère, personnalisation	100
repère, style	100
répétition courte	145
répétition de mesure	145
répétition et liaison de tenue	48
répétition, pourcent	145
répétition, utilisation de q	151, 321
répétitions	92, 134
réplique, fin	198
répliques	192
répliques, formatage	195
reprise avec alternative	134
reprise avec alternative et paroles	262
reprise avec levée	136
reprise courante	134
reprise de portée	178
reprise et anacrouse	136
reprise et liaison	140
reprise et liaison de prolongation	137
reprise et paroles	258
reprise manuelle	140
reprise, expansion	143
reprises	92, 134
reprises ambiguës	140
reprises avec alternatives et liaisons de prolongation	48
reprises développées	489

reprises et numéros de mesure	97, 139
reprises imbriquées	140
<code>resetRelativeOctave</code>	727
respiration, indication	123
<code>rest</code>	51
<code>rest-event</code>	194
<code>restrainOpenStrings</code>	320
retouche (tweak)	573
retouche des notes d'ornement	103
retour aux propriétés par défaut de la métrique	61
<code>retrograde</code>	13, 727
rétrograde, transformation	13
<code>revertTimeSignatureSettings</code>	727
<code>rfz</code>	112
rgb, couleur	205
<code>rgb-color</code>	205
<code>RhythmicStaff</code>	171
right aligning text	661
<code>right-align</code>	225
<code>right-margin</code>	502
<code>rightHandFinger</code>	357, 727
rôle	268
rôle, indication	279
rotating text	662
<code>rounded-box</code>	228
rvb, couleur	205
Rythme en MIDI	488
rythmique d'une mélodie	73

S

<code>s</code>	53
<code>sacredHarpHeads</code>	36
<code>sacredHarpHeadsMinor</code>	36
<code>SATB</code>	275
saut de ligne	510
saut de ligne et ligature	76
saut de page	537
sauts	124
sauts de durée	53
sauts de ligne	90
sauts de ligne et cadences	69
sauts de ligne et musique non mesurée	69
sauts de ligne réguliers	511
sauts de page et cadences	69
sauts de page et musique non mesurée	69
scalable vector graphics	484
<code>scaleDurations</code>	47, 69, 727
scaling markup	670
scaling text	662
Scheme, objet	693
Scheme, pure containers	600
Scheme, unpure containers	600
Scheme, variable	692
<code>score-markup-spacing</code>	501
<code>score-system-spacing</code>	501
<code>scoreTitleMarkup</code>	459
Scottish highland bagpipe	375
script et silence multimesure	55
seconde fois	134
segno	91, 101, 109, 687
segno sur une barre de mesure	216
sélection de la taille (notation)	200
septième	386
sesqui-bémol	8
sesqui-dièse	8
<code>set</code>	78
<code>set-octavation</code>	21
setting extent of text objects	684
setting horizontal text alignment	655
setting subscript in standard font size	645
setting superscript in standard font size	645
<code>settingsFrom</code>	727
<code>sf</code>	112
<code>sff</code>	112
<code>sfz</code>	112
<code>shape</code>	728
<code>shiftDurations</code>	728
<code>shiftOff</code>	158
<code>shiftOn</code>	158
<code>shiftOnn</code>	158
<code>shiftOnnn</code>	158
<code>short-indent</code>	189, 504
<code>show-available-fonts</code>	236
<code>showFirstLength</code>	483
<code>showFirstLength</code>	692
<code>showKeySignature</code>	375
<code>showLastLength</code>	483
<code>showLastLength</code>	692
<code>showStaffSwitch</code>	305
silence d'église	56
silence multimesure	51
silence multimesure et point d'orgue	55
silence multimesure, ajout de texte	55
silence multimesure, contraction	55
silence multimesure, étiquette	55
silence multimesure, expansion	55
silence multimesure, positionnement	56
silence multimesure, script	55
silence, spécification du positionnement vertical	51
silences	51
silences d'espacement	53
silences invisibles	53
silences multimesure et doigtés	58
silences multimesures	54
silences, collisions entre	58
silences, condenser les	58
silences, décalage automatique	158
silences, division	72
silences, mesure entière	54
simple closure	690
simple text strings	647
simple text strings with tie characters	675
simultanées, liaisons d'articulation	119
simultanées, liaisons de phrasé	122
<code>skip</code>	53, 728
<code>skipTypesetting</code>	483
<code>slashChordSeparator</code>	395
slashed digits	683
<code>slashedGrace</code>	728
<code>slur-event</code>	194
<code>slurDashed</code>	119
<code>slurDashPattern</code>	120, 728
<code>slurDotted</code>	119
<code>slurDown</code>	119
<code>slurHalfDashed</code>	120
<code>slurHalfSolid</code>	120
<code>slurNeutral</code>	119
<code>slurSolid</code>	119

slurUp	120
small	200, 225
smaller	223, 225
smob	693
snap pizzicato	316
snappizzicato	687
Solesmes	409
solo	163
son	485
sortie, définition	692
sos	307
sostenuto, pédale	307
sostenutoOff	307
sostenutoOn	307
soufflet	112
soufflet penché	592
soufflet, ligature	89
souligné	223
sourdine	109, 687
sous-ligature, orientation	83
southernHarmonyHeads	36
southernHarmonyHeadsMinor	36
sp	112
spacing	534
spacingTweaks	728
Span_stem_engraver	305
spécification d'un repère	100
spp	112
Sprechgesang	285
staccatissimo	109, 110, 687
staccato	109, 110, 687
stacking text in a column	651
Staff.midiInstrument	486
Staff_symbol_engraver	185
standard, taille des notes	201
start-repeat	140
startGroup	211
startStaff	178, 181
startTrillSpan	132
Stem	305
stem-spacing-correction	534
stemDown	207
stemLeftBeamCount	87
stemNeutral	207
stemRightBeamCount	87
stemUp	207
stencil	693
stencil, suppression	586
stopGroup	211
stopStaff	178, 181, 185
stopTrillSpan	132
storePredefinedDiagram	348
storePredefinedDiagram	728
stringTuning	332, 728
stringTunings	331, 344
style d'accidentelle	23
style d'altération <i>modern</i>	26
style d'altération <i>modern accidental</i>	26
style d'altération <i>neo-modern-cautionary</i>	27
style d'altération <i>neo-modern-voice</i>	28
style de repère	100
style de trait, liaison	119, 122
style de voix	158
styledNoteHeads	728
styles de métrique	59

styles, tête de note	33
sub	224
subdivision de ligature	82
subscript text	648
substitution de doigt	201
suggestAccidentals	417
super	224
superscript text	648
surimpression d'objets	587
sus	389
suspension de portée	178
sustain, pédale	307
sustain, style de pédale	308
sustainOff	307
sustainOn	307
SVG, format de sortie	484
syllabes, durée automatique	242
symbole arabe d'un demi-bémol	439
symbole de portée	178
symboles d'ornementation	109
symboles de portée, dessin	581
symboles de septième majeure	396
symboles non musicaux	230
synchronisation des notes d'ornement	106
syntaxe du mode markup	221
system-count	504
system-separator-markup	506
system-system-spacing	501
système	172
système choral	172
système pianistique	172
système, début de	172
système, grand	172
système, indicateur de séparation	177
systèmes imbriqués	176
systems-per-page	504

T

tabChordRepeats	728
tabChordRepetition	728
tablature	171, 317
tablature et glissando	324
tablature et indication d'harmoniques	324
tablature pour banjo	317
tablature pour guitare	317
tablature, accordages prédéfinis	331
tablature, clef	334
tablatures par défaut	319
tablatures personnalisées	331
tablatures pour banjo	362
tablatures, bases	319
TabStaff	171, 319
TabVoice	319
tag	477
tag	728
taille des notes	200
talon	687
taor	375
taqasim	441
teaching	28
teaching, style d'altérations	28
teeny	200, 225
tempo	63

temporel, quadrillage.....	208	time	58, 78, 728
temporelle, note de bas de page.....	466	times	42, 69, 728
temps, gestion du.....	108	timeSignatureFraction	69
ténor, clef.....	17	tiny	200, 225
tenue et nuances successives.....	113	tiré, indication d'archet.....	314
tenuto.....	109, 110, 687	tirer l'archet.....	687
tessiture.....	30	tiret, trait de liaison.....	119, 122
test de mesure.....	99	tirettes d'accordéon, symboles.....	308
tête de note allongée.....	39	tocItem	728
tête de note en losange.....	315	Top	1
tête de note et improvisation.....	39	top-level, texte.....	219
tête de note, Aiken.....	36	top-margin	499
tête de note, allure.....	36	top-markup-spacing	501
tête de note, apprentissage.....	34	top-system-spacing	501
tête de note, Christian Harmony.....	36	toplevel-bookparts.....	692
tête de note, Funk.....	36	toplevel-scores.....	692
tête de note, Harmonica Sacra.....	36	tracé d'objets graphiques.....	228
tête de note, harpe sacrée.....	36	trait d'union.....	250
tête de note, Walker.....	36	transcription de musique ancienne.....	175
têtes de note.....	200	transformation rétrograde.....	13
têtes de note, spéciales.....	33	translate	226
têtes de note, styles.....	158	translate-scaled	226
text	308	translating text.....	662
text columns, left-aligned.....	659	transparent, objet.....	587
text columns, right-aligned.....	661	transparentes, notes.....	204
texte ajouté.....	221	transpose	10, 12, 729
texte en colonnes.....	227	transposedCueDuring	198, 729
texte en préambule.....	219	transpositeur, instrument.....	10
texte et alternative.....	142	transposition.....	10
texte et extenseur.....	214	transposition	22, 192, 729
texte et rembourrage.....	229	transposition des hauteurs.....	10
texte hors marges.....	213	transposition et clef.....	17
texte indépendant.....	219	transposition et diagramme de fret.....	346
texte indépendant et note de bas de page.....	467	transposition et MIDI.....	22
texte indiquant le nombre de mesures vides.....	55	transposition, instrument.....	22
texte isolé.....	219	tre corde.....	307
texte justifié.....	227	treCorde	307
texte multiligne.....	227	tremolo	147
texte sur plusieurs lignes.....	227	tremolo, cross-staff.....	148
texte, alignement.....	225	trémolo, indication de.....	147
texte, alignement horizontal.....	225	trémolo, ligatures de.....	147
texte, alignement vertical.....	226	tremoloFlags	147
Texte, autres langues.....	212	triades.....	386
texte, centrage sur la page.....	227	triangle	230
texte, décoration.....	228	trill	132
texte, encadrement.....	228	trille.....	109, 687
texte, maintien dans les marges.....	213	trilles.....	132
texte, mise en forme des extenseurs.....	214	trilles avec hauteur explicite.....	133
texte, mise en forme des prolongations.....	214	trilles avec hauteur explicite et altération.....	133
texte, taille.....	223	trilles et MIDI.....	488
texte, top-level.....	219	trioletts.....	42
textLengthOff	213	trioletts, formatage.....	43
textLengthOn	56, 213	tupletDown	42
textLenthOff	56	tupletNeutral	42
textSpannerDown	214	TupletNumber	43
textSpannerNeutral	214	tupletNumberFormatFunction	43
textSpannerUp	214	tupletSpannerDuration	43
thumb	202	tupletUp	42
tieDashed	48	turc, noms des notes.....	443
tieDashPattern	728	turque, musique.....	443
tieDotted	48	tweak	729
tieDown	48	tweak (retouche, affinage).....	573
tieNeutral	48	tweak et points de contrôle.....	575
tieSolid	48	two-sided	503
tieUp	48	type	562

type de caractère 690

U

U.C. 307
 ukulele 336
 ukulele, tablature 331
 una corda 307
unaCorda 307
underline 223
 underlining text 650
 une pause par mesure 54
unfold 143
unfoldRepeats 729
unHideNotes 204
 Unicode 481
 unpure containers, Scheme 600
 UTF-8 480

V

varcoda 109, 687
 variables 450
 variables, utilisation de 475
 variante 181
 variante rythmique 266
 Vaticana, Editio 408, 409
VaticanaStaff 171
 VaticanaStaffContext 419
 VaticanaVoiceContext 419
 vents 372
 vérification d'octave 9
 vérification des limites de mesure 99
 vertical, espacement 518, 538
 vertical, positionnement des nuances 114
 vertically centering text 663
 visibilité d'objets 586
 visibilité d'un clef octaviée 591

visibilité des hampes 207
 vocalise 247
voice 23, 25
Voice 155
voice, style d'altérations 25
voiceOne 155
void 729
 voix 155
 voix entre deux portées 305
 voix multiples 158
 voix multiples et altérations 26, 28
 voix, \autoBeamOff et \partcombine 77
 voix, citation 195
 voix, décalage 158
 voix, division 278
 voix, réplcation 195
 voix, styles 158
 volta 134

W

Walker shape, tête de note 36
walkerHeads 36
walkerHeadsMinor 36
whichBar 94
with-color 205
withMusicProperty 729
wordwrap 227
wordwrap-lines 233

X

x11, couleur 206
 x11, couleurs 205
x11-color 205, 206
xNote 729
xNotesOn 729