

譜面を書く為の入門 -19 January 2019-

MUSIX TEX

Mu_{si}X_TE_X コマンド集

コマンドの解説は p_TE_X でのコンパイルを意図しての説明文.

吉田征夫

//参照 : musixdoc.dvi//

//参照 : 「やさしい 楽譜の読み方」 一成美堂出版 //

この小冊は $\text{T}_{\text{E}}\text{X}$ で楽譜を書く為の読み物です。メロディーと 1 程度の伴奏楽器を含めた譜面用を書くその足がかり的なもので、誰でも簡単に書けることの説明を意図したものです。もちろん、オーケストラ用の譜面も書く(写す)こともできますが、いきなりそういった専門へ挑戦することは考えていません。

譜面記述は $\text{pT}_{\text{E}}\text{X}$ と pdf での画面表示や印刷を目的で書いています。
(dvips,ps2pdf,acrobat 等々で dvi ファイルは ps,pdf ファイルへ可能です)

譜面書きは $\text{pT}_{\text{E}}\text{X}$ へ `musixtex.tex` というマクロパッケージを読み込んで実現します。`musixtex.tex` の版面仕様は A4 版組ですが、市販の一般的な譜面用紙 (A4) は文書用 A4 版より若干大きめです。五線譜行も市販のそれですと一ページ約 12 行程度ですが、`musixtex.tex` 仕様では $\text{T}_{\text{E}}\text{X}$ で書く場合には約 11 行程度で自然改頁されます、 $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X}2\epsilon$ では約 10 行程度で自然改頁されます。もちろんこれらは譜面書きの条件に左右されますが、

`musixtex.tex` を読み込んだ、 $\text{pT}_{\text{E}}\text{X}$ 、 $\text{pL}^{\text{A}}\text{T}_{\text{E}}\text{X}2\epsilon$ へ及ぼす、版面の変更は以下の値です (もちろん変更は可能です)。

```
\parindent = 0.0pt
\hoffset   = -43.81721pt
\voffset   = -28.45274pt
\hsize     = 540.60236pt
\vsiz      = 739.77165pt
```

例えば `jarticle` の値は、それぞれ、9.62216pt 0.0pt 0.0pt 327.0pt 1000.0pt ですので注意をしてください。

少し混乱するかもしれませんが、これらの pt は寸法の単位 *pt* で、大きさの pt ではありません。大きさの *pt* にはフォントがありますが、同じ単位で比較しやすいので **pt** で表現しました。もっとも、 $\text{T}_{\text{E}}\text{X}$ は最終的に頁 (版面) を大きさととらえ、頁版面を [幅/高さ/深さ] で処理します。

!) **MusiX $\text{T}_{\text{E}}\text{X}$** は $\text{T}_{\text{E}}\text{X}$ システムを用いて楽譜 (譜面) を取得できるプログラムです
 $\text{T}_{\text{E}}\text{X}$ 配布 URL の "W32 $\text{T}_{\text{E}}\text{X}$ based on $\text{T}_{\text{E}}\text{X}$ Live 2018 and Kpathsearch 6.3.0" —Full installation—
には組み込み済みです。

URL: <http://core.ring.gr.jp/pub/text/TeX/ptex-win32/current/win64> も download して 64bit-running 環境にする

!!) $\text{T}_{\text{E}}\text{X}$ システムにおいて、ほかに必要な running プログラムは
GPL Ghostscript 9.26 (現時点での最新の公開版, `gs926w64full-gpl.exe`)

URL: <http://core.ring.gr.jp/pub/text/TeX/ptex-win32/gs>
 $\text{T}_{\text{E}}\text{X}$, Ghostscript もインストール後にパソコンの環境設定 path に自分自身で記述しなければなりません、この作業は慎重に行ってください。ここは、プログラム running へ直に影響する記述です。

MusiX_TE_X の解説書, musixdoc.tex のコンパイル
\texmf-dist\doc\generic\MusiX_TE_X\ の中に同梱の musixdoc.tex を, そのディレクトリ内で
 コンパイルすると musixdoc.dvi を得ます. コンパイルは,
 >latex musixdoc.tex 実行,
 >mendex musixdoc 実行,
 >musixflx musixdoc 実行,
 >latex musixdoc.tex 実行,
 以上の計 4 回です, (dviout 表示での error のポップアップは「中止」をクリックしてください,
 musixdoc.dvi が表示印刷可能です.) この musixdoc が Musix_TE_X の解説書です.
 (musixdoc.dvi を)
 >dvips -P dl musixdoc.dvi 実行,
 >ps2pdf musixdoc.ps 実行で
 musixdoc.pdf(systemu 配布同梱のものと同じファイルを) 作成できます.
 musixdoc.tex のコンパイルは, 必ず latex.fmt の方でコンパイルして下さい .

pdflatex で running するなら,
 原稿ソースファイル”musixdoc.tex”の 30 行目の記述である.
 \usepackage{dvips,bookmarks=true,} の”dvips,”の文言を削除すれば一気に
 >pdflatex musixdoc.tex 実行で”musixdoc.pdf”を取得できます.

このドキュメントは p_TE_X 書式での解説になっています.
 (私の場合 p_TE_X 方が原稿記述が楽な理由からです)

!!!) p_TE_X 書式での MusiX_TE_X のコンパイル
 MusiX_TE_X そのものは 3 回のコンパイル (three pass system) 手順で DVI ファイルを得ます. この小
 冊での譜面記述は p_TE_X 仕様での記述を主たるとして説明です [もちろん, p_LA_TE_X2_ε 記述で platex
 running での譜面原稿は記述できます (此方が標準)]. 楽譜を書いたファイルのコンパイルは, 例
 えば sample.tex として次のような手順です.

コンソール画面 (通称 Win32 窓) で—拡張子は省略可—,

1. >ptex sample.tex
2. >musixflx sample
3. >ptex sample.tex

これで, sample.dvi が得られます. 後は

>dvips -P dl sample.dvi 実行,
 >ps2pdf sample.ps 実行で
 sample.pdf が作成され画面表示/印刷ができます.

再コンパイルの時は必ず >del *.mx?を実行してください.

始めに書いたように、メロディー (パート) とその伴奏楽器用 (パート) を含めた譜面用の実際のサンプル例です。

Deep River

Slowly with expression

from ビクター SF-5

コード記述は以下のとおりです。

```
\centerline{\Bigtype Deep River}
\instrumentnumber{2}
\setstaves 1{2}
\setstaves 2{1}
\setclef 1{6000}
\setclef 2{0000}
\generalsignature{-1}
\generalmeter{\meterfrac {4}{4}}
\startrule
\nobarnumbers
\sepbarrules
\startmuflex\OliveGreen\startpiece
%
% ● 1(小節毎に垂直に ; 1-3 が一番下行: 1-2 が中段行: 1-1 が一番上行,---の記述.)
% 1-3
\notes\ds|%
% 1-2
\ccharnote M{\ppffsixteen mp}\ca c&%
% 1-1
\Uptext{\medtype\bf Slowly with expression}\ds\en
\doublebar
%
% ● 2
% 2-3
\notes\zqp M\ibu0K0\qbp0I\zq M\tbbu0\tbu0\qb0H\zq M\fl H\qu H%
\zq K\qu G\sk\zq G\qu C|%
% 2-2
\icresc\zqp d\ibu0d1\qbp0b\zq f\tbbu0\tbu0\qb0c%
\zcharnote M{\tcresc}\zq{fd}\na b\qb0b%
\fl b\zql b\ibu0f{-1}\qbp0f\tbbu0\tbu0\qb0d%
\icresc\ust l\rq L\rql b\ibu0c1\qb0c\ust l\tbu0\qb0d\zcharnote K{\tdecresc}&%
% 2-1
\sk\sk\hsk\pause\en
\bar
%
% ● 3
% 3-3
\notes\ust a\zq J\qu F\ust a\tie Ju1\zq J\tie Fd1\qu F%
\fermataup N\zh J\hu F|%
% 3-2
\ust l\zq f\qu a\ust l\tie fu1\zq f\tie ad1\hu a%
\fermataup l\zh f\hu a%
% 3-1
\sk\pause\en
```

```

\doublebar
% ● 4
% 4-3
\notes\islurd 0F\ibu0H4\qb0{FJM}\tbu0\qb0a\ibl0M{-2}\qb0c%
\tslur 0H\tbl0\qb0N\ql M|%
% 4-2
\ccharnote b{\ppffsixteen mp}\itieu 0h\itied 1f\icresc%
\zh h\hu f\sk\zcharnote a{\tcresc}\zcharnote a{\ppffsixteen legato}%
\sk\sk\ttie 0\zq h%
\ttie 1\ibu0h{-1}\qb0f\zq g%
\zcharnote a{\decrecendo{10mm}}\tbu0\qb0e\zq f\qu c&%
% 4-1
\ibsluru 0h\itied 1h\zcharnote a{\smalltype Deep \Rule}%
\hu h\sk\sk\sk\ttie 1\ibu0i{-1}\qb0h\tbu0\qb0g\tbsluru 0f\qu f\en
\bar
% ● 5
% 5-3
\notes\hup I\sk\fl K\ql K|%
% 5-2
\icresc\arpeggio N4\zq g\zq d\qu b\zcharnote K{\tcresc}\zh{db}\hu N%
\zcharnote K\icresc%
\zq f\na b\zq b\qu M\zcharnote K{\tdecrec}&%
% 5-1
\zcharnote a{\smalltype riv -}\qu g%
\zcharnote a{\smalltype er,}\hu d\zcharnote a{\smalltype my}\qu f\en
\endpiece\par
\rightline{{\it from} ビクター SF--5}

```

1 小節目の説明にある如く、パートの垂直記述で進行が Musix \TeX の基本書式です。殆どコマンドが占めていますが、これらのコマンドは MusixItem.txt のメニューを呼び出して書いていけます。ここでのサンプル記述例は、一小節/一小節を `\notes.....\en` の中で書いていますが、いろいろな記述し方があります。一音一音や数音を纏めて `\notes.....\en` で括って書く事も出来ます。但し、 \TeX の `{.....}` のようなネスト構造は持っていないので注意してください。コマンドの意味をある程度理解していると、このコード記述は読めるようになります、つまり、記述できるようになります。これらコマンドは覚えるのではなく、ある程度のパターン認識で十分です。忘れてならないのは書式記述は”垂直記述で進行する”ことです。

上記例を、もう少し丁寧に既要しますと(混み入らないでください、いずれ理解すれば非常に簡単明快な事ですので)、1 個のメロディーライン (A) とピアノ伴奏 (B) 譜です。A と B のパート区切りは”&”で明示し B パートの中の Trebel と Bass の区切りは”|”で行っています。例えば書き出しは `\notes` 等々で始めて、垂直に全てのパートを書き終えたら `\en` で閉じ、次の小節を再び `\notes` 等々で書き始める..... こんな感じです。書式の方法さえ理解できれば、コマンド等々は譜面を書いていれば自ずと備わってきます。

ちなみに、この MusiX \TeX の入力をユーザーがもっと簡単に入力出来るようにプロプロッサ (前処理) として M^P_P を開発していたのが、Jan Nieuwenhuizen 氏で (LilyPond にはその多くの特徴が受け継がれています)。

お使いの Editor でコンソール画面 (cmd 窓) でコンパイル制御関係をお使いの事でしょうが、通常 cmd 窓でコンパイルを実行する時に F-7 キーを押して見てください、実行したコマンド文字列が line 番号順に小窓に表示が出ると思います。—こちらを使ってのコンパイルも作業はし易いです。

また、譜面書きは慣れるまで一小節書いてはコンパイルをし結果を確認しながら書き進めていく方が早道です。譜面書きが初めてであればなおさらです。必ず、記述ミスでコンパイルは止まります。始めから一気に譜面コンパイルが完結すると思わないでください。

1) 五線譜の環境設定コマンド.

TeX は音楽記号等々を独自に決めていません. 音楽界で決められている記号等々を TeX が出力するだけです.

関連ファイルの呼び込み

• `\input colordvi.tex`

TeX(pTeX) での color を利用する為.

• `\input musixtex.tex`

musixtex.tex は譜面用のマクロ集で, 譜面用の独自の版面設定をデフォルトで持っています.

—参照:ii page—

概ね musixdoc.dvi の紹介順番どおりに書いていきます.

1) 五線譜の用意

• `\instrumentnumber{n}`

n 用意する五線譜 (パート) の数を指定. メロディーだけなら, 五線譜パートは 1 つでよいでしょうけど, ピアノやギター伴奏をも書くなら, それらを同時に一緒に書くために設定します. 自動改行/自動改頁 (強制改行/強制改頁 OK) ですので, なにかまわず譜面を書き進めていく事ができます. また, 最終的に楽譜が耽美的な仕上げかどうかは, 書く人の技量に左右されます.

• `\setstaves n{N}`

n 番目パート自身がさらに五線譜パートを N 数 (デフォルト: 最大 4 パート) 持つ指定.

2) 音部記号の指定

• `\setclef n{S1S2S3S4}`

n 番目の N 個のパートの各々音部記号を数字で指定.

S₁ S₂ S₃ S₄ の 4 つの連パートに記述する数値の音部記号の位置は,

数値 = 音部記号の意味

0 = ト音記号, デフォルト値.

1~ 4 = ハ音記号, 五線譜の第 1 線上~ 第 4 線上.

5 = ヘ音記号, 五線譜の第 3 線上.

6~ 7 = ヘ音記号, 五線譜の第 4 線上~ 第 5 線上

9 = ト音記号, 五線譜の第 1 線上.

8 is not used

N 個の連パートが例えば 1 つのヘ音記号パートしかない場合は残りの 3 個のパートは 0 を書き, 必ず {6000} 等と全てを埋めて指定しなければなりません.—参照:iv page—

ト音記号 = バイオリン, フルート, クラリネット, サックス, トランペット, ホルン, ギター, 等々の楽譜に用いられる記号で, もっとも一般的な記号.

ヘ音記号 = チェロ, コントラバス, ファゴット, トロンボーン, チューバ, エレキベース, 男性パート (合唱の), 等々の楽譜に使われる記号.

ハ音記号 = ビオラ (アルト譜表), チェロ, トロンボーン, 第 2 線はメゾソプラノ譜表, 第 4 線はテノール譜表, 等々に使われる記号.

MusixTeX では, 複数のパートがある場合は常に最初の記述順番から一番下の順に五線譜パートが出力されます, 音部記号 (clef) もそれに倣っています, S₁ もその連パートの一番下の段で s₄ がその一番上のパート出力になっています.—参照:iv page—

3) 調号の指定

• `\generalsignature{S}`

全パートの調号の設定, S = 数値で長音階/短音階に関係なく, S が S > 0 の数値なら, その数値に見合う sharp (#) 記号を楽譜の決まりに則った出力をします,

S が S < 0 の数値なら, その数字に見合う flat (b) 記号を楽譜の決まりに則った出力をします.—参照:iv page—もちろん, 小節内で音符の変調号もその線上に書けます.

• `\setsign n{S}`

n 番目のパートの S 調号, S = 数値は上に同じです.

4) 拍子の設定

• `\generalmeter{\meterfra{ }{ }{ }}`

全パートの拍子の設定.

$\frac{3}{4}$ 調子なら, 単に `\meterfrac{3}{4}` とも記述できます, その場合は $\frac{16}{4}$ 調子なら, `\meterfrac{16}{4}` と記述になります. $\frac{4}{4}$ 調子なら `\meterC` と書けます—参照:iv page—, $\frac{2}{2}$ 調子なら `\allabreve` と書けます.

- `\setmeter n{m}`

n 番目のパートの m 拍子設定, `\setmeter 2{\meterfrac{16}{4}}` なら, 2 番目のパートは 4 分の 16 拍子の記述です.

5) (パート) に名前を付ける.

- `\setname_n{ 楽器の名前等 }`

n 番目のパートである五線譜の左側にそのパートの楽器名等々を出力する.

6) 五線譜左端線の有無.

- `\nostartrule`

五線譜の左端線無し. 五線譜の左端線を引きたい時は, このコマンドの記述をない. デフォルトでは左端線があります.

- `\startrule`

五線譜の左端線有り. デフォルトの値で, 通常省略しても左端線を引きます.

6) 小節番号の制御.

- `\nobarnumbers`

小節番号無し. 小節番号に関するコマンド記述がない場合は, デフォルトで小節毎にその小節線左上に番号を出力していきます.

- `\systemnumbers`

五線譜行毎に, 五線譜行頭の左上に枠付きでその小節番号のみ出力する. MusiX_{TeX} の `musixdoc.dvi` には, さらに, 小節番号の出力位置を変えるマクロ記述も書かれています.

- `\startbarno=n`

小節番号を n 番から始める.

7 五線譜の行末線の制御.

- `\sepbarrules`

各パートごとに独立した右端線 (改行の時の) をひく. デフォルトでは複数あるパートの行末線は各パートにまたがってひかれます.

此処までの, 五線譜環境のコマンド記述順列挙を纏めます.

```

\input colordvi.tex%   color 譜面を書くため.
\input musixtex.tex%   楽譜環境の読み込み.
\instrumentnumber{n}%   パート数.
\setstaves n{p}%       n 番目のパートの p 連パート.
\setclef n{s1s2s3s4}%   n 番目の p 連パートの音部記号
\generalsignature{s}%   全パートの調号.
\setsignn{s}%          n 番目のパートの調号.
\generalmeter{m}%      全パートの拍子
\setmeter n{m}%        n 番目のパートの拍子
\setnamen{ 楽器の名前 }%   n 番目のパートの左側へ出力する.
\nostartrule%          五線譜の左閉線は出力しない.
\startrule%            左閉線を出力する, 省略可.
\nobarnumbers%         デフォルトの小節番号を出力させない.
\systemnumbers%        小節番号出力のバリエーション.
\startbarno=n%         小節番号 n 番から出力させる.
\sepbarrules           複数パートの時の行末線を各パート独立してひく.

```

これらの属性を途中で変えて出力させる事が出来ます.—参照:xxvi page—

- もっと簡単な記述. 1パートのみで単に楽譜を書いていくのなら.

```
\input musixtex.tex
```

```
\startpiece
```

楽譜を書いていく

```
\endtpiece
```

```
\bye%%これを, 例えば example.tex と save して, 後は TeX でコンパイルする. これで取得できます.
```

これだけで OK です. 音部記号や調号や拍子 は `startpiece` の上の行へ追加挿入していけば OK です.

`startpiece` の説明は次ページ以後にあります.

此処までのコマンド記述が, 五線譜の環境コマンド記述です. 以下からは, この属性を持った五線譜 (五線) 上に音符を書いていくコマンド列挙に入ります.

五線譜上の音符・音楽記号等のコマンド

ここでの楽譜関係の決まりは音楽界の様式で決まっており、MusixTeX や TeX 等々で決められている物ではありません。MusixTeX は、それらの決まり(記号等々)をコマンドで出力しているだけです。

1) • \startmuflex\startpiece

前ページ一連の設定後に、実際の音符関連の記述開始をする宣言です。 \startpiece 前に譜面関係のコマンドが何も記述が無く、いきなり、 \startpiece から始めますと、ト音記号で五線譜を出力します。

\startmuflex は version T.39 以後の版では省略出来るとかいてあります。その場合は \startpiece の中で自動対応し(コンパイル画面に \startmuflex launched by \startpiece の表示あり)ます。

2) • \endpiece

譜面記述の終了宣言です。 \startpiece と対になっています。 \startpiece と \endpiece の間に譜面を書いていきます。このコマンドには次のようにバリエーションがあります。

楽譜の全終了時のコマンド。

<code>\Endpiece</code>	細い線/太い線の二線出力で全終了宣言
<code>\Stoppiece</code>	上に同じ
<code>\endpiece</code>	普通の小節線出力で全終了宣言
<code>\setdoublebar\endpiece</code>	細い二重線で全終了宣言
<code>\setdoubleBAR\endpiece</code>	太い二重線で全終了宣言
<code>\setrightrepeat\endpiece</code>	反復終了と同時に全終了宣言
<code>\stoppiece</code>	上に同じ
<code>\zstoppiece</code>	小節線無しで全終了宣言

これらのコマンドは、 \startpiece で始まった譜面書きを全て終える意思表示(終了宣言)の働きをします。

もちろん、 pTeX のコンパイル終了指定である \bye をファイルの最終行に記述する事を忘れないでください。

3) • Pitches(音程)

音の高低位置(スケール)をアルファベット大文字小文字で表しています。


The image shows two musical staves. The top staff is a treble clef staff with notes for letters a through z. The bottom staff is a bass clef staff with notes for letters A through Z and a through e. The notes are represented by red circles on the staff lines.

音程の位置とその音程の記号の一覧表です。音程位置は通常では音程を表す文字アルファベットで書いていきます。また、音程の位置は数字での表記方法もあります。


4) • Notes(音符)


音符は音の高低(既記のスケール表示)と、下記のような音の長さを表わしており、音符コマンドはこの二つを同時に持つコマンドです。?に既記の音程位置のアルファベット記号を書きます

$\text{♩} = \backslash\text{hu} ?$ 2分音符上向き	$\text{♩} = \backslash\text{hup} ?$ 符点2分音符上向き
$\text{♮} = \backslash\text{hl} ?$ 2分音符下向き	$\text{♮} = \backslash\text{hlp} ?$ 符点2分音符下向き
$\text{♫} = \backslash\text{qu} ?$ 4分音符上向き	$\text{♫} = \backslash\text{qup} ?$ 符点4分音符上向き
$\text{♬} = \backslash\text{ql} ?$ 4分音符下向き	$\text{♬} = \backslash\text{qlp} ?$ 符点4分音符下向き
$\text{♩} = \backslash\text{cu} ?$ 8分音符上向き	$\text{♩} = \backslash\text{cup} ?$ 符点8分音符上向き

 = `\cl ?` 8分音符下向き

 = `\clp ?` 符点8分音符下向き

 = `\ccu ?` 16分音符上向き

 = `\ccl ?` 16分音符下向き

● デフォルトでは符点16音符コマンドはありません。
別ファイルを読み込めば、OKになります。



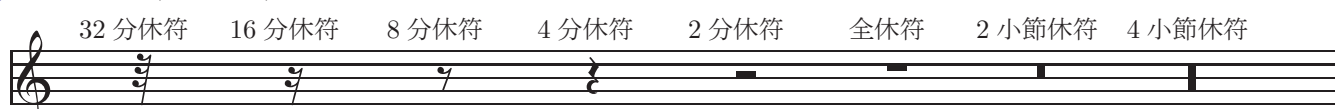
`\hu i` `\hup i` `\hl j` `\hlp j` `\qu i` `\qup i` `\ql j` `\qlp j`

`\cu i` `\cup i` `\cl j` `\clp j` `\ccu i` `\ccl j` `\zw i`

`\zw ?` = 全音符 (符尾 = 棒が付きません). u = 符尾 (棒) が上向き, l = 符尾 (棒) が下向き. u, l の代わりに **a** を用いると符尾は上/下に自動判断で付けられます. `\ha i`, `\qa j`, `\ca i``\cca j` 等々.

5) • Rests (休符記号)

32分休符 16分休符 8分休符 4分休符 2分休符 全休符 2小節休符 4小節休符



`\hs` `\qs` `\ds` `\qp` `\hpause` `\pause` `\PAuse` `\PAUSE`


断っておきますが、これは音楽 (符) の休息の意味ではありません、メロディの中でのもっとも大切な“間”のことで、役者が舞台で見せる“間”と同じことです。

休符記号の出力位置を `\raise n\Interligne` を使って上下に移動させる事ができます (n = 五線幅 n 本分)。

`\raise-2\Interligne\qp` と記述すれば、デフォルト (音楽界の決まり) 出力より2線分下へ出力できます。

但し、`\pause`, `\hpause` は、上下移動の出力に `\liftpause {n}` と `\lifthpause {n}` と単独の専用コマンドを使います (かならず中カッコを忘れず書いてください、 n の単位は上記と同じです)。

6) • Accidentals (臨時記号)




`\csh g` `\cfl h` `\cna i` `\cdf l j` `\cdsh k` `\sh g` `\fl h` `\na i` `\df l j` `\dsh k`
or `^g` or `_h` or `=i` or `>j` or `<k`

`\uppersh 1` `\upperna m` `\upperfl 1`


臨時記号出力は大きさを大 (`\bigaccid`) と、小 (`\smallaccid`) の二つ持っておりデフォルト出力は自動判断されて出力されます。どちらか一方のみを出力させたいなら、`\startpiece` の前に `\bigaccid` と宣言します。あるいは、音符へ個別に指定したいなら `\bigsh` と `\smallsh` 等と記述します。上記の例はデフォルト任せの出力です。

臨時記号出力位置を左へずらしたいなら、`l` を使って例えば、`\lsh` と `\lfl` 等と記述すれば OK です。

7) • Accents (アクセントの記号)



`\lpz g` `\upz j` `\lsf g` `\usf j` `\lst g` `\ust j`



`\lppz g` `\uppz j` `\lsfz g` `\usfz j` `\lpzst g` `\upzst j`

8) • Non-spacing(chord)notes

9) • Shifted non-spacing(chord)notes

様々な和音を構成する時に (符頭) だけで長さを持たない音符記号を出力させます。

`\zh p` (p = 二分音符)

`\rh p` 符頭を右にずらす。

`\lh p` 符頭を左にずらす

`\zq p` (p = 四分音符)

`\rq p` 符頭を右にずらす。

`\lq p` 符頭を左にずらす

`\rw p` 全音符 p 符頭を右にずらす。

`\lw p` 全音符 p 符頭を左にずらす

triad(三和音) の例 :

The image displays two musical staves. The first staff shows a sequence of triads: C, D, E, F, G, A, B, C. The second staff shows a sequence of suspended fourth chords: Csus4, Dsus4, Esus4, Fsus4, Gsus4, Asus4, Bsus4, Csus4. Each chord is represented by a set of notes on a five-line staff.

和音とはハーモニー (= コード) のことで、2つ以上の高さの異なる音が同時になったときの響きです。三和音は三つを積み重ねたものを言い、上記の例はハ長調の主音(ド)を根音にしたドレミファソラシドの順番で作ったものです。Cは通常C majorと言っているコードです。Csus4はC suspended 4thといい、ポピュラーミュージックで良く使われるコードです。

上例の記述は、Dは `\notes\zh{d^f}\nh h\en`

Bは `\notes\zh i\lsh k\zh k\sh m\nh m\en`

Csus4は `\notes\zq c\rq g\lq f\en`

`\notes.....\en` 等々の説明は次ページ以後に有ります。このように音符等々コマンドは `notes` と `en` に挟んだ中で書いていきます。さらに、`\zh{d^f}` のように音程は違うが同じ音符群は中カッコで一括に記述する事も出来ます。

10) • Single non-spacing notes

一つの棒に複数の符頭をつけた音符を出力します。

書くべき普通の音符コマンドの前に `z` をつけて書き、最後にそれら符頭が付けられるべき通常の音符を書きます。また、`z` の代わりに、`r` を付けると右にずれ、`l` を付けると左へずれた音符を出力します。

The image shows a musical staff with a single note. The note has multiple stems and flags, illustrating the use of the `\notes` command with multiple stems.

上例の記述は

`\notes\rq h\ibu 0k0\zqb 0{gi}\qb0 k\rq h\zqb 0{gi}\qb0 k%`

`\rq h\zqb 0{gi}\qb0 k\rq h\tbu 0\zqb 0{gi}\qb0 k\en`

`\bar`

`\notes\arpeggio c5\tie ful\tie hul\tie kul\zh{fh}\hu k\zh{fh}\hu k\en`

`\ibu`(ビームの開始), `\tbu`(ビームの終了), `\tie`(弧線), 等々の説明も次ページ以後に有ります。このように音符に付随するコマンド等々も `notes` と `en` に挟んだ音符の直前へ書きます。

`\arpeggio pn` p = 音程, n = またぐ五線の数。この記号は和音を同時でなく連続奏法の意味です。

譜面を書くと言うのは、`\startpiece` で始まり、音符を現すコマンド、記号を現すコマンド等々を `\notes.....\en` の中に挟んで書き綴り、最後に `\endpiece` で譜面記述を終了し、このファイルを `\bye` のコマンドで終わりを告げ、ファイルに名前を付けて閉じれば譜面書きは終了です。後は、このファイルを pTeX でコンパイルすれば譜面の電子表示情報や紙面印刷譜面と実現できます。

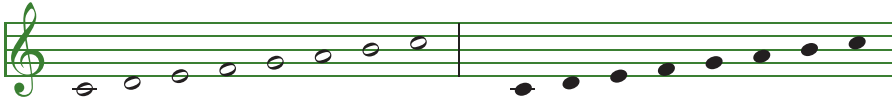
音符はたった約 12 個程度しか有りません、この 12 個の言葉が多様に変化するのが音楽と言う曲の世界です。こんな世界は他には有りません、日本語は何万語もあり、そのようなもの覚えて文章を書く世界では有りません。

11) • Single(spaceing)stemless notes

棒がない音符を書くときのコマンドです.

`\nh p` ($p =$ 二分音符)

`\nq p` ($p =$ 四分音符)



これも記述は :

`\notes\nh{cdefghij}\en`

`\bar`

`\notes\nq{cdefghij}\en`

12) • Spacing of the notes(空白を扱うコマンド)

全て `\notes.....\en` の中で用います.

`\sk` 1\noteskip の空白.

`\hsk` 0.5\noteskip の空白.

`\bsk` 1\noteskip の空白 (左へ戻る).

`\qsk` 音符の頭 1 つ分の空白.

`\hqsk` 音符の頭半個分の空白.

`\off{D}` D の長さの空白.(単位は \noteskip です. 例 `\off{4\noteskip}`)

左右に位置をずらして出力する (長さはありません).

`\roff{note}` 音符の頭一個分の右に出力.

`\loff{note}` 音符の頭一個分の左に出力.

pTeX の glue コマンド `\hskip` 数値 pt や `\vskip` 数値 pt や `\kern` 数値 pt も空白として使えますが, 此方のコマンドは `musixtex.tex` の属性を一切持っていませんので使用には十分注意してください.

13) • Easy selecting notes spacing(音符と音符のスペース)

音符と音符をスペースで繋いでいくコマンドです。









• `\notes\ql l\ql k...&\cu f.....|\ccu b.....&\hu f.....\en`

.....&.....&..... は複数パートがある場合各パートの区切りです、

&.....|.....&..... はそのパートの更なる連パートの区切りです。

つまり、音符とそれに付随する音楽記号等々は `\notes.....\en` に挟んで書くか、あるいは `\notes.....\en` の中に書いていきます。複数パートの&.....&記述で注意しなければならないのは、一番最初の&記述が一番下の段として出力されていくこと事です。これは、連パート|..... でも同じです。---参照:iv page---

`\notes.....\en` には、いろいろなバリエーションが用意されています。すべては、音符と音符間のスペースの長さ(寸法)の違いです。`\notes... \en` が基本で、その寸法の何倍かでバリエーションのコマンドがあり、これらのスペースは、音符の持っている長さ(時間)と擬似されています。(音符の持っている長さ(時間)を、寸法の空白に置き換えるわけです。)

usage	spacing	suggested use for
<code>\znotes ...&... ...&... \en</code>	(non spacing)	specials
<code>\notes ...&... ...&... \en</code> (<code>\notesp...&... ...&... \en</code>	2\elemskip 2.5\elemskip	 = 16th 16th pointed, 8th triplet) 別途読み込み
<code>\Notes ...&... ...&... \en</code>	3\elemskip	 = 8th
<code>\Notesp ...&... ...&... \en</code>	3.5\elemskip	 = 8th pointed, 4th triplet
<code>\NOtes ...&... ...&... \en</code>	4\elemskip	 = 4th
<code>\NOtesp...&... ...&... \en</code>	4.5\elemskip	 = 4th pointed, 2th triplet
<code>\NOTes ...&... ...&... \en</code>	5\elemskip	 = 2th
<code>\NOTesp...&... ...&... \en</code>	5.5\elemskip	 = 2th pointed
<code>\NOTEs ...&... ...&... \en</code>	6\elemskip	 = 1th

`\normalmusicsize : elmskip=6pt`(デフォルト値) `\smallmusicsize : elmskip=4.8pt`

必ずしも、`\notes.....\en` 中に 16 分音符のみしか置けないのではなく、8 分音符や 4 分音符なども実際には置いて譜面を書いていきますし、理想から言えば、その音符はその音符の長さに類似したスペースを持つ、コマンドの中に置いて譜面を書いていくでしょうが、それですべてが美しい譜面仕上げになるとは限りません。

`\def\NOTes{\vnotes4.3\elemskip}` と `\startpiece` の前に記述して spacing を変える事も出来ます。

14) • `\bar`

小節を区切るコマンドです。`\bar` コマンドは、五線をまたぐ縦線を引(出力)きます。また、様々なバリエーションがあります。

<code>\bar</code>	細い小節線。(自動改行/改頁を認める)
<code>\zbar</code>	小節線なし。(自動改行/改頁を認める。)
<code>\xbar</code>	小節線あり。(改行を認めない。)
<code>\doublebar</code>	細い二重小節線
<code>\setdoublebar</code>	細い二重小節線へ変更させる
<code>\setdoubleBAR</code>	太い二重小節線へ変更させる
<code>\setemptybar</code>	小節線なし

改行と改頁 (`\bar` に付随する) のコマンド。

`\alaligne` 小節線付き, (強制) 人意的改行指示です。

`\alapage` 小節線付き, (強制) 人意的改頁指示です。

`\zalaligne` 小節線なし, (強制) 人意的改行指示です。

`\zalapage` 小節線なし, (強制) 人意的改頁指示です。

細い二重線で改行させたいなら、`\setdoublebar\alaligne` と記述します。

自然改行で `color` の情報が次頁最初の一行へ反映されない場合、**自然改頁されたその行で `\endpiece\supereject` と人為改頁を行い、次頁以後で `\Blue\startpiece` 等と再度色指定をして譜面書きスタート続行してください。** `color` で書いていないならこの操作は必要ありません、自然改頁のまま続行でよろしいです。

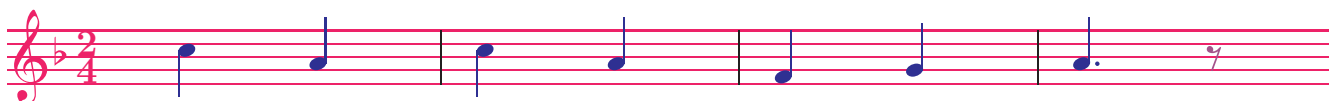
ここまでで、簡単な譜面が書けます。一応サンプルとして4小節のを載せてみます。color で書いていますが、それについては最終頁 named-color 一覧表と譜面を参照してください。

ソースコードは

```
\input colordvi.tex
\input musixtex.tex
\instrumentnumber1
\setclef1\treble
\setsign1{-1}
\generalmeter{\meterfrac24}
\nostartrule
\nobarnumbers
\Largemusicsize
\WildStrawberry\startpiece
%% 1
\NOTes\hsk\Blue{\qa{jh}}\en%
\bar
%% 2
\NOTes\Blue{\qa{jh}}\en%
\bar
%% 3
\NOTes\Blue{\qa{fg}}\en%
\bar
%% 4
\NOTes\Blue{\qup h}\DarkOrchid{\ds}\en%
\endpiece
\bye
```

コンパイルすると次のように出力します。

題「つき」の一部分です。—作詩・作曲者 不詳—



2拍子ですから、リズムは「強」「弱」の繰り返しです。

また、3小節目のような音符部分は `\qu{fg}\en%` とも記述できます。

五線譜サイズを始めから制御できます。

`\smallmusicsize 16pt`

`\normalmusicsize 20pt` デフォルト

`\largemusicsize 24pt`

`\Largemusicsize 29pt`

これらを、`\startpiec` の前で指定します。

複数パートの一つのパート五線譜のみを変更するなら、

`\setsize n{SIZE}` で可能です。SIZE = `\tinyvalue` `\smallvalue` `\largevalue` `\Largvalue`

pTeX で譜面関係のドキュメント文書 (論文等々) を書くなら、1パートで書いた五線譜属性はデフォルトへ戻して `\endpiece` ください。例えば、`\setsign1{0}`、`\normalmusicsize` 等々です。楽譜用譜面 **One-only** なら関係ありません。

15) • Beames (ビーム).

連桁記号の音符です.



符尾の付いた連続音符を略して連桁記号で記述することができます. 連桁記号はデフォルトで4本引けます. 別途ファイル読み込みで, さらに増設して引く事は可能です. 斜めの線も $-9 \sim 9$ の範囲で調節自由です,

beam の開始宣言と終了.

`\ibu nps.....\tbun\qbnp` *upper beam*

u : 上符尾音符 n : 参照値 (0~4 まで). p / p : pitch(開始/終了音程) s : [-9~9] の値

`\ibl nps.....\tbln\qbnp` *lower beam*

l : 下符尾音符 パラメータの意味は同じ (same parameter meaning)

`\ibbu nps.....\tbun\qbnp` *double uper beam*

u : 上符尾音符 n : 参照値 (0~4 まで). p / p : pitch(開始/終了音程) s : [-9~9] の値

`\ibbl nps.....\tbln\qbnp` *double lower beam*

l : 下符尾音符 パラメータの意味は同じ (same parameter meaning)

`\ibbbu nps.....\tbun\qbnp` *triple upper beam*

u : 上符尾音符 n : 参照値 (0~4 まで). p / p : pitch(開始/終了音程) s : [-9~9] の値

`\ibbbbl nps.....\tbln\qbnp` *triple lower beam*

l : 下符尾音符 パラメータの意味は同じ (same parameter meaning)

`\ibbbbu nps.....\tbun\qbnp` *quadruple upper beam*

u : 上符尾音符 n : 参照値 (0~4 まで). p / p : pitch(開始/終了音程) s : [-9~9] の値

`\ibbbbl nps.....\tbln\qbnp` *quadruple lower beam*

l : 下符尾音符 パラメータの意味は同じ (same parameter meaning)

連桁ビーム内の音符記号は, 上符尾・下符尾に関係なく `\qbnp(n= 参照値, p= 音程)` と書き進めていきます.

複雑なビーム表現のサンプル.

サンプルのコード記述からその意味を取ってください.



`\N0tes\ibbbu0e2\qb0e\tbbbu0\qb0f\tbbu0\qb0g\tbu0\qb0h\en`

`\ibbbu0`= 上符尾ビーム3本を参照値0と名づけて描く, `\tbbbu0`= 参照値0の3本目のビーム終了,

`\tbbu0`= 参照値0の2本目のビーム終了, `\tbu0`= 参照値0のビーム終了.



`\N0tes\ibl0m{-2}\qb0m\nbbl0\qb0l\nbbbl0\qb0k\tbl0\qb0j\en`

`\ibl0`= 下符尾ビームを参照値0として描く, `\nbbl0`= 参照値0へ下符尾ビーム1本追加する (合計2本),

`\nbbbl0`= 参照値0へ下符尾ビーム1本追加する (合計3本),

`\tbl0`= 参照値0のビーム終了.



```
\N0tes\ibu0e2\qb0e\tbbu0\tbu0\qb0g\en
```

\ibu0= 上符尾ビームを参照値 0 として描く, \tbbu0\tbu0= 参照値 0 へ上符尾ビームを 2 本描いて終了.



```
\N0tes\ibl0\roff{\tbb10}\qb0m\tbl0\qb0l\en
```

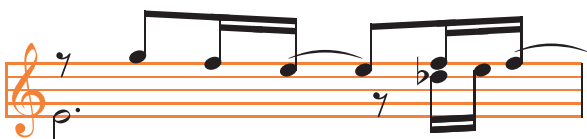
\ibl0= 下符尾ビームを参照値 0 として描く, \roff{\tbb10}= 参照値 0 へ下符尾ビームを 1 本追加して, 追加分は此処で終了.\tbl0= 参照値 0 のビーム終了.



```
\N0tes\ibbbu0f2\roff{\tbbbu0}\qb0f\tbbu0\qbp0g\tbu0\qb0h\en
```

```
\N0tes\ibbl0k{-2}\roff{\tbb10}\qbp0k\qb0j\tbbb10\tbb10\tbl0\qb0i\en
```

説明は, 省略します.



下側のビーム部分記述は `\fl k\ibbl 0k1\qb0k\loff{\loff{\tbl0\qb0l}}` です.

その他の多彩なビーム表現方法については, musixdoc.dvi を参照してください (26 頁~ 31 頁).

Beames には連符を意味する出力の場合もあります. 連符とは通常一音符を 2 等分するところを 3 で割ったり, 3 等分するところを 2 や 4 で割った音符や休符の事です. が, 実際にはこのルールに則さないで書かれている譜面もありまので. 演奏する時に注意を要します.

次の例で, 最初の連符は 4 分音符を 3 で割った例です, 最後の連符は符点 4 分音符を 4 で割った例です.



```
\notes\def\txt{\eightit 3}\downtrio a20\ibu0f0\qb0{ff}\tbu0\qb0f\en
```

```
\notes\def\txt{\eightit 4}\uptrio p30\ibbl0j0\qb0{jjj}\tbl0\qb0j\en
```

`\downtrio pns` `\uptrio pns` (p= 音程, n=n\noteskip, s= ブラケットの傾き) で, ブラケットの中央に `\txt` を出力する.

Brackets and parentheses

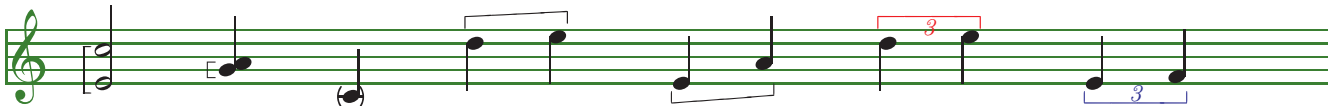
```
\bracket{p}{n} p = 音程, n = 縦幅 (n\noteskip)
```

```
\doublethumd{p} p = 音程.
```

```
\lpar{p}, \rpar{p} p = 音程.
```

```
\ovbkt{p}{n}{s}, \unbkt{p}{n}{s} p = 音程, n = 横幅 (n\noteskip) s = 傾き
```

`\uptrio`, `\downtrio` は上記参照.



上から順番に出力例です.

```
\notes\bracket e5\zh j\hu e\doublethumd g\rq h\qu g\lpar c\rpar c\qu c%
```

```
\ovbkt o12\q1{k1}\unbkt b13\qu{eh}\uptrio o10\q1{k1}\downtrio b10\qu{ef}\en
```

16) • **Ties and Slurs**(タイとスラー).

タイやスラーは音をつなげたり,なめらかに演奏したりするのを指示する記号です.

tie(タイ) は同じ音の高さの音符を結んだ弧線のことです.

tie の開始と終了.

上符頭に付ける.

`\itieu np.....\ttie n` n = 参照値 (0~5 まで), p = 音程

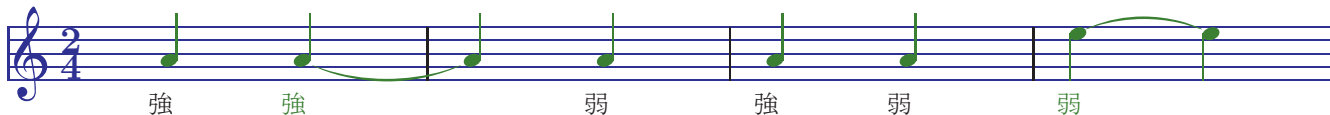
下符頭に付ける.

`\itied np.....\ttie n` n = 参照値 (0~5 まで), p = 音程



```
\Notes\itied 0e\qu e\ttie 0\qu e\en
\Notes\itieu 0l\ql 1\ttie 0\ql 1\
```

従って, 条件によっては tie がつけられる事によって, リズムの強拍が前に移動する, syncopation(シンコペーション)が発生します.



本来, 2 拍子ですから「強」「弱」で進行する譜面ですが, tie がつけられた事によって, 拍が前にずれ込み syncopation が発生しています. 演奏する時に要注意です.

slur(スラー) は異なる高さの音符を結んだ弧線のことです.

slur の開始と終了.

上符頭に付ける.

`\isluru np.....\tslur np` n = 参照値 (0~5 まで), p/p = 開始/終了音程

下符頭に付ける.

`\islurd np.....\tslur np` n = 参照値 (0~5 まで), p/p = 開始/終了音程



```
\notes\islurd 0c\qu{cdf}\tslur 0g\qu g\en
\notes\isluru 0h\ql{hij}\tslur 0k\ql k\en
```

スラー間の音を管楽器系ではタンキングせずに演奏します, 弦楽器系では弓の切り返しをせずに演奏します.

タイとスラーのバリエーション.

コード記述より, その意味を受け取ってください.



```
\notes\ibsluru 0c\qu{cdf}\tbsluru 0g\qu g\en
\notes\ibslurd 0h\ql{hij}\tbslurd 0k\ql k\en
```

(棒の上/下につけた弧線です). このコマンドを使うとき, 下記の **tie**, **slur** との同時使用は避けてください弧線が符頭/符尾の弧線に設定変更されます.

tie や **slur** の長さがわかっている時.

`\slur p1p2sl`

$p1$ = 開始の音程, $p2$ = 終了の音程, s = u(上付き) / d(下付き) l = 長さ (単位は \noteskip).

`\tie psl`

p = 音程, s = u(上付き) / d(下付き) l = 長さ (単位は \noteskip).

tie や slur の弧線のふくらみの指定.

`\midslur{h}` (tie や slur を閉じるコマンドの直前に置く)
h= 弧線の頂点の高さ, 単位は \internote(五線の幅の半分), デフォルトは 3 です.

tie や slur の弧線の傾きの指定.

`\curve hij` (tie や slur を閉じるコマンドの直前に置く)
h=\midslur(最大垂直距離) i= h までの距離 j= h までの距離. デフォルトは \curve 344 です.

tie や slur での改行前後.

デフォルト設定では, 改行前部は tie とみなされ, 改行後部は slur の開始音程と同じです.

前部を変えるには, \breakslur np(n= スラー参照番号, p= スラーのきれる音程)

後部を変えるには, \liftslur np(n= スラー参照番号, p= スラーの開始位置, 単位は \internote) です.

- tie や slur の弧線の dotted での出力.



```
\notes\dotted\ibsluru 0c\qu{cdf}\tblsluru 0g\qu g\en
\notes\dotted\isluru 0h\ql{hij}\tsslur 0k\ql k\en
```

その他に, 複数連パート間をまたいで slur 弧線を引く事ができます. 詳細は musixdoc.dvi(38 頁~ 45 頁) を参照してください.

連符のブラケット出力以外に, slur による連符記号も出力できます.



```
\notes\slur ffd2\xtuplet 3a\ibu0f0\qb0{ff}\tbu0\qb0f\en
\notes\slur jju3\xtuplet 4o\ibbl0j0\qb0{jjj}\tbl0\qb0j\en
\xtuplet np n= 連符数, p= 連符数の出力位置.
```

他に, \triolet, \butext, \bltext, 等々ありますが, 連符 slur の詳細は musixdoc.dvi の索引で, 列記したコマンドの参照をお願いします.

17) • Small and tiny notes(装飾音符).

装飾音符は元の音符を華やかにする為の飾り付けで, 作曲された時代や演奏される時代によって奏法が異なる事があります. 装飾音符は小さい音符で書き, その小節内の拍子数には含まれないです. 装飾音符のサイズには tinysize と smallsize があり, \notes.....\en の中で装飾音を付けたい音符の直前(前打音)に記述します.

```
{\multnoteskip\smallvalue\smallnotesize p} p = 音符
{\multnoteskip\tinyvalue\tinynotesize p} p = 音符
```



必ずしも slur(弧線)を書く決まりはありません. 第3小節の slur は必要です, 後打音と言って元の音符は前の音符です. コード記述(一小節目), 後続もこれに習い記述なので省略します.

```
\notes\qu j\multnoteskip\tinyvalue\tinynotesize%
\ibbu 0k{-2}\islurd 0k\qb0 k\tbu 0\qb0 j\en
\Notes\tslur 0j\qu j\ql {lj}\en
\bar
( 三小節目)
\Notes\islurd 0j\ql j%
\tslur 0i\multnoteskip\tinyvalue\tinynotesize\grcu i\en
\notes\loff{\ql j}\qp\hsk\en
\bar
```

装飾音符が八分音符なら \grcu p, \grcl p 等々で斜線の入った装飾八分音符を出力できます(上記の例で3小節目の装飾音符).

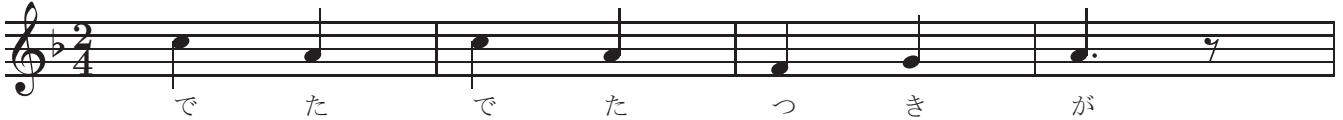
譜面の中の文字列出力とフォント

音楽記号の数々

18) • Putting anything anywhere (譜面の中へ文字列, 音楽記号等の出力).

`\zcharnote p{text}` p の音程位置に text を右方向へ出力します。
`\lcharnote p{text}` p の音程位置に text を左方向へ出力します。
`\ccharnote p{text}` p の音程位置に text を中央へ出力します。
`\uptext{text}` 五線の上で text を右方向へ出力します。
`\Uptext{text}` 五線の2本分上に text を右方向へ出力します。

text の部分へ歌詞や音楽記号等々を入れて譜面を書きます。 長さが無いので影響はありません。



このソースコードは次のように書きました。

```
\startpiece
% 1
\NOTes\zcharnote M{で}\ql j\zcharnote M{た}\qu h\en
\bar
% 2
\NOTes\zcharnote M{で}\ql j\zcharnote M{た}\qu h\en
\bar
% 3
\NOTes\zcharnote M{つ}\qu f\zcharnote M{き}\qu g\en
\bar
% 4
\NOTes\zcharnote M{が}\qup h\ds\en
\endpiece\par
```

簡単な置き換えマクロの話,

上記の沢山な `\zcharnote M{...}` のコマンドを, 打ち込んでいくのも, また, 後から音程 M の位置を変えたいときや, 歌詞のフォントをもっと小さくしたい時など, この記述では不便なので, 次のようなマクロを作り `\startpiece` より上の行に置いておきますと, 歌詞付き譜面書きも安易に進行できます。

`\def\SONG#1{\zcharnote M{#1}}` SONG は任意の name です。

上記のソースコードは以下のように安易に記述できます (記述の為の小節番号の%コメントは省いてあります)。

```
\def\SONG#1{\zcharnote M{#1}}
\startpiece
\NOTes\SONG{で}\ql j\SONG{た}\qu h\en
\bar
\NOTes\SONG{で}\ql j\SONG{た}\qu h\en
\bar
\NOTes\SONG{つ}\qu f\SONG{き}\qu g\en
\bar
\NOTes\SONG{が}\qup h\ds\en
\endpiece\par
```

`\def\SONG#1{\zcharnote M{#1}}` の M の音程を変えれば譜面の全歌詞 (SONG) 部分の位置が応じて変化します。また, 最後の #1 を `{\zcharnote M{\Mineight #1}}` とすれば, 譜面の歌詞 (SONG) 部分の全てを `\Mineight` のサイズにする事ができます。フォントについては, 次ページ以後で説明してあります。

歌詞専用の高度なコマンドも `musixtex.tex` には備えられてありますので, `musixdoc.dvi` を必ず参照してください。ここで紹介しているのは, 簡単な譜面を書いていく為の限定した私流なものですから。

注意 `\def\name` における, 任意な name の名付けですが, 必ずアルファベット大文字で名付けてください。理由は, `TEX` をコアにしたシステムでは特殊な例外を除いて, システムのコマンド名はすべて小文字を用いられています, 大文字列で書くことによって, このコマンドが私用コマンドであることを, ファイルを見た第三者に識別できるように明示できる為です。

例えば, 貴方の書いた譜面を何かの理由で印刷屋さんへ渡すような時に備えての親切心です。

19) • musixtex.tex 中でのフォント.

MusiXtex.tex の専用フォント[¶]

Fonts

Size/pt	style	example
8pt	<code>\smalltype</code>	small roman
8pt	<code>\smalltype\bf</code>	small bold
8pt	<code>\smalltype\it</code>	<i>small italic</i>
10pt	<code>\normtype</code>	normalroman
10pt	<code>\normtype\bf</code>	normal bold
10pt	<code>\normtype\it</code>	<i>normal italic</i>
12pt	<code>\medtype</code>	medium roman
12pt	<code>\medtype\bf</code>	medium bold
12pt	<code>\medtype\it</code>	<i>medium italic</i>
14pt	<code>\bigtype</code>	big bold
17pt	<code>\Bigtype</code>	Big bold
20pt	<code>\BIGtype</code>	BIG bold
25pt	<code>\BIGtype</code>	BIG bold
16pt	<code>\ppffsixteen</code>	<i>pp, ff, poco a poco</i>
20pt	<code>\ppfftenty</code>	<i>pp, ff, Pastorale</i>
24pt	<code>\ppfftentyfour</code>	<i>pp, ff, crescendo</i>

和語フォントについて.

pTeX でのソース記述ですので和語フォント変更は自分で作るのみです. フォントの命名とサイズの拡大縮小一般形は次の記述です.

`\font\name=` フォント名 `scaled` 数値 あるいは `\font\name=` フォント名 `at` 寸法 pt
(`name` は任意な名前です).

24pt のゴシックフォントを作るなら,

`\font\GothTwntyfour=goth10 scaled 2400` あるいは `\font\GothTwntyfour=goth10 at 24pt`

8pt の明朝体フォントを作るなら,

`\font\Mineight=min10 scaled 800` あるいは `\font\Mineight=min10 at 8pt`

pTeX で予約されている和語フォント名もあります.

`\tenmin(=min10)`, `\sevenmin(=min7)`, `\fivemin(=min5)`

`\tengt(=goth10)`, `\seventgt(=goth7)`, `\fivegt(=goth5)`

これらは, そのコマンド名で直ぐにそのサイズのフォントを使えます. 他に縦組用和語もありますが省きます.

[¶] Type1(pfb) のフォントも用意されています, Windows OS 用になりますが.

20) • **Metronome**(メトロノーム記号の出力).

メトロノームは音楽の細かな速度を表し、楽譜の最初の部分に記述される演奏の進行速度のことです。

`\metron{\qu}{60}` ⇒ ♩ = 60 等と出力します。

♩ = 92 ~ 96



`\notes\Uptext{\metron{\qu}{\midtype 92 \sim 96}}\en`

.....

これは、一分間に四分音符を 92~96 回打つ速度を意味しています。普通の歩く速さが Andante で四分音符で約 72 回打つ速度です。

速度を表す音楽言葉.

<i>Grave</i>	約 ♩ = 40 以上	重々しく
<i>Largo</i>	約 ♩ = 46 以上	幅広く緩やかに
<i>Lent</i>	約 ♩ = 50 以上	気楽に
<i>Adagio</i>	約 ♩ = 58 以上	静かに緩やかに
<i>Andante</i>	約 ♩ = 72 以上	歩く速さで
<i>Andantion</i>	約 ♩ = 80 以上	やや早足程度で
<i>Moderato</i>	約 ♩ = 92 以上	中くらいの速さで
<i>Allegretto</i>	約 ♩ = 108 以上	快速に/マーチ
<i>Allegro</i>	約 ♩ = 130 以上	快速に
<i>Vivace</i>	約 ♩ = 160 以上	活発に速く
<i>Presto</i>	約 ♩ = 180 以上	急速に速く

速度を変化させる音楽言葉.

<i>rit.</i> (ritardando)	だんだん遅く
<i>rall.</i> (rallentando)	だんだん緩やかに
<i>riten.</i> (ritenuto)	平均的に遅く
<i>accel</i>	だんだん速く
<i>Tempo rubato</i>	好きな速さで
<i>a tempo</i>	変えた速さを (rit,rall,accel...), 変える前の速さに戻って
<i>Tempo l(Tempo primo)</i>	変えた全体の速度を (Allegro,Andante...), 最初の速度に戻って

速度標語に添える言葉.

<i>assi</i>	きわめて	ジャズ, ポップスなどでよく使うリズム標語
<i>molto</i>	きわめて	<i>Medium 4 Beat</i> 中くらいの 4 拍子で
<i>meno</i>	より少し	<i>Medium fast</i> 中くらいよりやや速く
<i>poco a poco</i>	少し	<i>Slow</i> ゆっくりと
<i>poco</i>	少しずつ	<i>Ballad</i> ゆっくりと
<i>un poco</i>	すこし	<i>Rubato</i> 自由の速さで
<i>non troppo</i>	はなはだしくなく	<i>Bossa</i> ボサノバのリズムで
<i>piu</i>	より多く	<i>Samba</i> サンバのリズムで
<i>e</i>	そして, ... と ...	<i>Afro Cuba</i> アフロ・キューバンリズムで
<i>sempre</i>	常に	

発想標語.

<i>Agitato</i>	せき込んで (激して)	<i>Animato</i>	元気に	<i>Alla Marica</i>	行進曲風に
<i>Amabile</i>	愛らしく	<i>Apassionato</i>	熱情的に	<i>Brillante</i>	華やかに
<i>Cantabile</i>	歌うように	<i>Comodo</i>	気楽に	<i>Con Brio</i>	いきいきと
<i>Dolce</i>	やわらかに	<i>Espressivo</i>	表情豊かに	<i>Grazioso</i>	優雅に
<i>Legato</i>	なめらかに	<i>Leggiero</i>	軽快に	<i>Maestoso</i>	厳かに
<i>Marcato</i>	はっきりと	<i>Pastorale</i>	田園風に		

21) • Intensity conventional signs(強弱記号出力コマンド).

`\pppp` \Rightarrow *pppp* `\ppp` \Rightarrow *ppp* `\pp` \Rightarrow *pp* `\p` \Rightarrow *p*
`\mezzopiano` ¶ \Rightarrow *mp* `\mf` \Rightarrow *mf* `\f` \Rightarrow *f* `\fp` \Rightarrow *fp*
`\sF` \Rightarrow *sf* `\ff` \Rightarrow *ff* `\fff` \Rightarrow *fff* `\ffff` \Rightarrow *ffff*

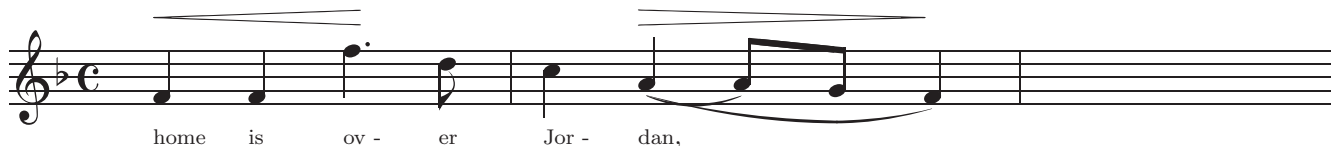
22) • Dynamic signs(強弱の変化を表す記号).

`crescend`(クレッシェンド)= だんだん強く, `decrescend`(デクレッシェンド)= だんだん弱く.

`\crescendo{l}` , `\decrescendo{l}` l = 単位 *mm*(最大 68mm まで).

あるいは,

`\icrsec` で開始宣言, `\tcresc` or `\tdcresc` で終了宣言する.



`\def\SONG#1{\zcharnote M{\smalltype #1}}%歌詞部分はマクロ記述にしました.`

`\startpiece`

`% 1`

`\Notes\zcharnote r{\crescendo{28mm}}\SONG{home}\qu f%`

`\SONG{is}\qu f\SONG{ov -}\qlp m\en`

`\Notes\SONG{er}\cl k\en`

`\bar`

`% 2`

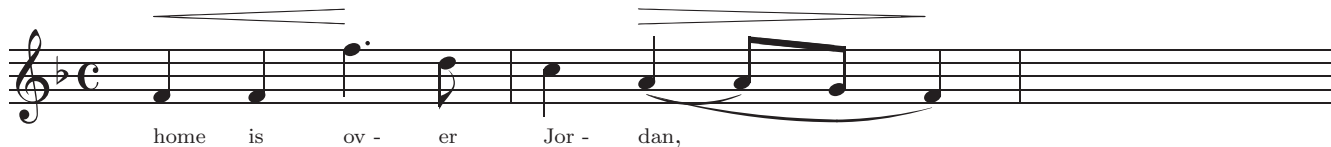
`\Notes\SONG{Jor -}\ql j\zcharnote r{\decrescendo{38mm}}%`

`\SONG{dan,}\islurd0h\islurd1h\qu h%`

`\tslur 0h\ibu 3h{-2}\qb 3h\tbu 3\qb 3g\tslur 1f\qu f\en`

`\bar`

`\endpiece`



`% 1` 上と同じですが, 歌詞部分をマクロにせず直書きにしてみました.

`\Notes\icresc\zcharnote M{\smalltype home}\qu f%`

`\zcharnote M{\smalltype is}\qu f\zcharnote r{\tcresc}%`

`\zcharnote M{\smalltype ov -}\qlp m\en`

`\Notes\zcharnote M{\smalltype er}\cl k\en`

`\bar`

`% 2`

`\Notes\zcharnote M{\smalltype Jor -}\ql j\icresc%`

`\zcharnote M{\smalltype dan,}\islurd0h\islurd1h\qu h%`

`\tslur 0h\ibu 3h{-2}\qb 3h\tbu 3\qb 3g\zcharnote r{\tdcresc}%`

`\tslur 1f\qu f\en`

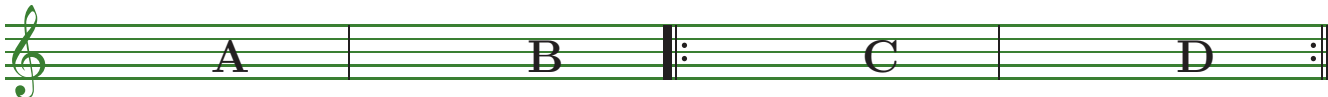
`\bar`

¶ musixdoc.tex の 61 頁に `\mp` がありますが, これは, `\mezzopiano` の方を使ってください, T_EX の二項演算子(数式記号)の呼び出しで既に `\mp` が定義されています. musixtex.tex の定義では `\mezzopiano` になっています.

23) • Repeats(リピート記号)

一定区間の小節を繰り返し演奏する為の記号です, 作曲者が同じことを書く手間の省略ができます.

`\leftrepeat` 反復を始める小節
`\alaligne\leftrepeat` 改行後反復の始まり小節
`\startpiece\leftrepeat` 譜面の開始から, いきなり反復の始まる小節
`\rightrepeat` 反復の終了小節
`\setrightrepeat\alaligne` 反復の終了小節で改行
`\setrightrepeat\endpiece` 反復の終了小節で譜面書き全終了
`\lefrightrepeat` 終わり始めの小節.



奏法 : A ⇒ B ⇒ C ⇒ D ⇒ C ⇒ D

コード記述は(上記の例は便宜上です, 音楽記号等々を必要個所にどのように記述しているかの一例のためにです),

```
\notes A(譜面を書いていく) \en
\bar
\notes B(譜面を書いていく) \en
\leftrepeat
\notes C(譜面を書いていく) \en
\bar
\notes D(譜面を書いていく) \en
\setrightrepeat
\endpiece
```

Repeats(リピート記号)のバリエーション

- カッコを用いる.

`\setvoltabox{text}` 反復(カッコの)始まる直前の`\bar`の前に記述
`\setendvoltabox` 反復終了小節の`\bar`直前に記述



奏法 : A ⇒ B ⇒ C ⇒ D ⇒ A ⇒ B ⇒ C ⇒ E

コード記述は(上記の例は便宜上です, 音楽記号等々を必要個所にどのように記述しているかの一例のためにです),

```
\notes A(譜面を書いていく) \en
\bar
\notes B(譜面を書いていく) \en
\bar
\notes C(譜面を書いていく) \en
\setvoltabox{1}\bar
\notes D(譜面を書いていく) \en
\setvoltabox{2}\setendvoltabox\rightrepeat
\notes E(譜面を書いていく) \en
\setendvoltabox\Endpiece
```

カッコの右閉じを嫌なら,`\setvoltabox`の代わりに,

`\setvolta{text}` 反復(カッコの)始まる直前の`\bar`の前に記述
(この場合は対応する閉じのコマンドは有りませんので記述なしでOK)

`\def\raisevolta{7\internote}`と`\startpiece`の前に記述しておけば, カッコ記号の出力位置を7\internote分上
にずらし上げて出力できます.

個別にカッコの出力位置上下を制御するには,`\notes.....\en`の直前に上記の変更記述を書けばOKですが, そのさい
, 上下位置変更宣言後に既にも実施されている`\setvoltabox`等の閉じコマンドを書くと不整になりますので十分に注意
してください.`\Setvolta`については,musixdoc.tex(musixdoc.dvi)を参照してください.

楽譜のリピートはリピート記号や上記のカッコを用いた, かつ, 次項以後の表示記号等々で複雑に絡み合わせた譜面になっ
ています. どれが優先順位が先かは私にはわかりかねます—通常は, リピート記号/カッコ/次項以後の表記記号/
の順位だと思いが—.

- **bis**(ビス), **ter**(テル), **quarter**(カテル) を用いる。
一定の区間の小節を繰り返す時に使います。bis=2回, ter=3回, quarter=4回です。

奏法 : A ⇒ B ⇒ C ⇒ B ⇒ C ⇒ D

コード記述は(上記の例は便宜上です, 音楽記号等々を必要個所にどのように記述しているかの一例のためにです),

```
\notes A(譜面を書いていく) \en
\setvoltabox{\ppffsixteen bis}\bar
\notes B(譜面を書いていく) \en
\bar
\notes C(譜面を書いていく) \en
\setendvoltabox\bar
\notes D(譜面を書いていく) \en
\Endpiece
```

bis の代わりに, *ter* なら ⇒ 奏法 : A ⇒ B ⇒ C ⇒ B ⇒ C ⇒ B ⇒ C ⇒ D
quarter なら B ⇒ C が 4 回奏されます。

\ppffsixteen という書体でなくとも, 好きな書体で OK です。また, カッコの垂直位置はデフォルトで 4\internote です。この位置を変えたいなら, 例えば \def\raisevolts{5\internote} 等と \raisevolts の値を変更すれば OK です。

- **Da Capo**(ダ・カーポ), **Fine**(フィーネ), **^** [Fermata (フェルマータ)] を用いる。

奏法 : A ⇒ B ⇒ C ⇒ D ⇒ A ⇒ B

D.C. = **Da Capo** の略。 **Fermata** = `\Fermataup p` or `\fermataup p` *p* = 音程
^ と *Fine* に挟まれる小節線は必ず細二重線 `\doublebar` で記述する決まりです(ここで, 曲は終了の意味)。
解説 : ダ・カーポ (*D.C.*) が付いているところから曲の一番最初に戻り, フィーネ (*Fine*) で曲は終わる。

コード記述は(上記の例は便宜上です, 音楽記号等々を必要個所にどのように記述しているかの一例のためにです),

```
\notes A(譜面を書いていく) \en
\bar
\notes B(譜面を書いていく)%
\ccharnote a{\ppffsixteen Fine}\Fermataup m\en
\doublebar
\notes C(譜面を書いていく) \en
\bar
\notes D(譜面を書いていく)%
\ccharnote a{\ppffsixteen D.C.}\en
\setdoublebar\endpiece
```

\ccharnote a 等々はここでの記述で, 実際にはそれに則しての記述にしてください。もちろん, \ppffsixteen も好きな書体指定で OK です。また, \Fermataup は \fermataup でも OK です。

Fermata Mark は音符に付けられて, その音符を奏者の好みの時間的長さに奏でる意味を持っていますので, `\Fermatadown` や `\fermatadown` というバリエーションも有ります。

- **Dal Segno**(ダル・セーニョ), **Fine**(フィーネ), **§** [Segno mark(セーニョマーク)] を用いる。

奏法 : A ⇒ B ⇒ C ⇒ D ⇒ B

D.S. = **Dal Segno** の略。 **Segno mark** = `\segno p` *p* = 音程


曲の終わりの決まりは概記に同じです。

解説：ダル・セーニョ(D.S.)が付いているところから、マークまで戻り、演奏を続けてフィーネ (Fine) で曲の演奏が終わります。

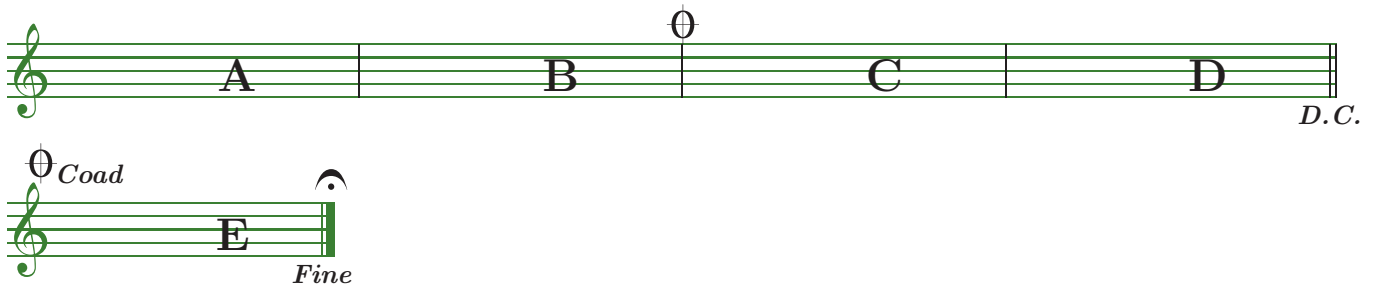
コード記述は(上記の例は便宜上です、音楽記号等々を必要個所にどのように記述しているかの一例のためにです)、

```
\notes A(譜面を書いていく) \segno m\en
\bar
\notes B(譜面を書いていく)%
\ccharnote a{\ppffsixteen Fine}\Fermataup m\en
\doublebar
\notes C(譜面を書いていく) \en
\bar
\notes D(譜面を書いていく)%
\ccharnote a{\ppffsixteen D.S.}\en
\setdoublebar\endpiece
```

記号や文字の出力垂直位置等々はここでの例です。実際にはそれに則しての値指定です。


`\Segno` と書くと  の記号出力になりますので注意してください。

- Coda(コーダ)を用いる。



奏法：A ⇒ B ⇒ C ⇒ D ⇒ A ⇒ B ⇒ E

Coda mark = `\coda p` p = 音程

 *Coad* が付いた譜面 **E** は別途譜面にて書かれます。上記の例は参照説明表示で実際の譜面で **E** も一小節で終了なんてありえません。

解説：ダ・カーポ(D.C.)が付いているところから曲の一番最初に戻り、コーダのマーク以後は別途記述譜面(Coad 譜面)を演奏して終了。

コード記述は(上記の例は便宜上です、音楽記号等々を必要個所にどのように記述しているかの一例のためにです)、

```
\notes A(譜面を書いていく) \en
\bar
\notes B(譜面を書いていく) \en
\bar
\notes\coda m\en
\notes C(譜面を書いていく) \en
\bar
\notes D(譜面を書いていく)%
\ccharnote a{\ppffsixteen D.C.}\en
\setdoublebar\endpiece
%% coda 用別途譜面
\startpiece
\notes\coda p\lcharnote p{\ppffsixteen Coad}\en%
\notes E(譜面を書いていく) %
\ccharnote a{\ppffsixteen Fine}\Fermataup m\en
\Endpiece
```

`\Coad` と書くと  の記号が出力しますので注意してください。

24) • Changing score attributes(属性の変更)

音部の変更

- `\setclef n{s1s2s3s4}`
小節内での変更 `\changeClefs`(長さあり)
`\zchangeClefs`(長さなし)

調号の変更

- `\generalSignature{s}`
- `\setSign n{s}`
小節内での変更 `\changeSignature`

拍子の変更

- `\generalMeter{m}`
- `\setMeter n{m}`
小節内での変更は無し。

小節後の変更は、`\bar` の代わりに下記のコマンドを使う、

- `\changeContext` 普通の小節線を引いた後に表示
- `\ChangeContext` 2本の小節線を引いた後に表示
- `\zchangeContext` 小節線なしで表示

それぞれのコマンドで変更値を設定し直後に変更コマンド名を記述すると、変更された記号が出力されます。



上記例の記述は、

```

\generalMeter\meterC
\generalSignature{-2}
\nostartrule
\startpiece
\notes\qu{ceg}\ql j\en
\generalMeter{\meterfrac{3\meterplus2}{8}}%
\changeContext%
\setclefs1{5000}%
\zchangeClefs%
\notes\hu J\hl c\en
\setclefs1{0000}%
\zchangeClefs%
\generalSignature{2}%
\changeSignature%
\generalMeter{\meterfrac{3}{4}}%
\notes\hsk\en
\ChangeContext%
\notes\ibbl0i0\qb0{hhh}\tbl0\qb0h\bsk\bsk\bsk%
\ibbu0i0\qb0{jjj}\tbu0\qb0j\en
\notes\hu h\en
\generalMeter\meterC%
\changeContext%
\generalSignature{-2}%
\changeSignature%
\notes\qu{ceg}\ql j\en
\endpiece

```

値を変更するコマンド, それを実施するコマンド, この二対のコマンドの直後には、必ず%を付けて改行等々をしてください。これは必須な絶対に必要な記述方法です。

68colors(CMYK model)
Pre-defined in T_EX
 colordvi.tex

GreenYellow		Yellow		Goldenrod	
Dandelion		Apricot		Peach	
Melon		YellowOrange		Orange	
BurntOrange		Bittersweet		RedOrange	
Mahogany		Maroon		BrickRed	
Red		OrangeRed		RubineRed	
WildStrawberry		Salmon		CarnationPink	
Magenta		violetRed		Rhodamine	
Mulberry		RedViolet		Fuchsia	
Lavender		Thistle		Orchid	
DarkOrchid		Purple		Plum	
Violet		RoyalPurple		BlueViolet	
Periwinkle		CadetBlue		CornflowerBlue	
MidnightBlue		NavyBlue		RoyalBlue	
Blue		Cerulean		Cyan	
ProcessBlue		SkyBlue		Turquoise	
TealBlue		Aquamarine		BlueGreen	
Emerald		JungleGreen		SeaGreen	
Green		ForestGreen		PineGreen	
LimeGreen		YellowGreen		SpringGreen	
OliveGreen		RawSienna		Sepia	
Brown		Tan		Gray	
Black		White			

カラーの楽譜と言うか譜面ですが下記にサンプルを載せました. 記述の仕方については, このファイルのソース最終行を参照してください. 一般的には, カラーにしたい部分を中括弧で囲むのですが一概にそう言い切れません. 中括弧で囲む事でエラーになる場合も生じます. ここで.....参照した記述は一例の記述です. 人によってカラーコマンドを使う位置が, それ故に, 人なりになると思います. 言葉で説明するよりもソースを見た方が認識は速いです. p_LA_TE_X2_ε 使用ならカラーコマンドの記述法はもう少し p_TE_X とは差異 (複雑) が生じると思います.

