

# hakase.sty 使用法

高吉 清文

ver.0.1 [March 2, 2019]

## 1 はじめに

`hakase.sty` は南山進流声明仮譜を描画するために作成された  $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ / $\text{u}\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$  及び  $\text{lua}\text{L}\text{A}\text{T}\text{E}\text{X}$  用のスタイルファイルである。譜の形は岩原諦信師の昭和改版進流魚山薑芥集に準拠した。

## 2 使用法

### 2.1 依存するパッケージ

このパッケージは主に縦書きで使用することを想定している。プリアンブルで以下を読み込むこと。

- `graphicx.sty`  
図形を扱う為に必要。
- `hakase.sty`  
パッケージ本体。
- `tikz.sty`  
`\usetikzlibrary{decorations.pathmorphing}`  
譜の作図はこのパッケージとライブラリーを利用。  
パッケージ本体で読み込まれる。
- `pxeveryshi.sty`  
`tikz` を縦書きで使えるようにするパッケージ。  
 $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ / $\text{u}\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$  に必要、 $\text{lua}\text{L}\text{A}\text{T}\text{E}\text{X}$  には不要。  
パッケージ本体で読み込まれる。

### 2.2 座標、描画単位

直交座標は紙面に対して水平方向を  $x$  方向とし右向きを正、垂直方向を  $y$  方向とし上向きを正とする。極座標の角度は  $x$  軸正方向を  $0$  度とし原点を中心に左回りに増加する。描画単位 `\tanni` の初期値は未定である。

座標の表現は以下の表式を用いる。

直交座標: (  $x$  座標 ,  $y$  座標 ) カンマで区切る

極座標 : ( 角度 : 長さ ) コロンで区切る

### 2.3 描画命令

---

`\hakase`


`\hakase{ <音符命令> }`

`\hakase[ <音符の単位長> ]{ <音符命令> }`

<音符の単位長>: 規定値に対する倍率 例) `\hakase[0.8]{\kak}`

声明の譜を描く。

例) これが呂のユリ `\tanni=1em\hakase{\yuriL{\c}}` です。

これが呂のユリ  です。

`\karifu`  
`\gyscale`

`\karifu(〈音符右移動〉, 〈全体上移動〉)[〈tbc〉]{〈文字列〉}{〈音符命令〉}`  
`\karifu[ ~ ]{ ~ }{ ~ } -> \karifu(0,0)[ ~ ]{ ~ }{ ~ }`  
`\karifu( ~ ){ ~ }{ ~ } -> \karifu( ~ )[ c ]{ ~ }{ ~ }`  
`\karifu{ ~ }{ ~ } -> \karifu(0,0)[ c ]{ ~ }{ ~ }`

〈tbc〉 : 文字と譜の位置揃え、t:上揃え、b:下揃え、c:中央揃え  
 〈音符右移動〉: 音符のみ右に移動、左は負値。文字間隔の調整に用いる。  
 〈全体上移動〉: 文字と音符全体が上に移動、下は負値。親文字に対する相対サイズで指定。  
`\gyscale` : 文字に対する相対サイズ。描画単位`\tanni=\gyscale × 〈文字サイズ〉`。  
 : 初期値は`\gyscale=0.5`である。

縦書きでは、文字の左横に声明仮譜を描く、横書きでは文字の下に描く。

例) 呂の`\karifu{ユリ}{\yuriL{c}}`

呂のユリ

例) 呂の`\def\gyscale{1}\karifu{ユリ}{\yuriL{c}}`

呂のユリ

`\Karifu`

グループ仮符

`\Karifu[〈tbc リスト〉]{〈文字列リスト〉}{〈音符命令リスト〉}`  
`\Karifu{〈文字列リスト〉}{〈音符命令リスト〉}`  
 各リストはコンマ区切り。

### 3 音符命令

この節の命令は前節の`\hakase`、`\karifu`の中で用いる。

#### 3.1 音階と基本音符

音階、音符表

五音宮商角徴羽は角度を使って表1のように表す。  
 45度で等分されていないのは見た目を原本に近付けるためとスペースの節約のためである。各音階を表す基本音符は音階の角度の方向に引いた直線である。今のところ三重の徴は定義していない。

表 1. 音階表

音階名	角度	命令
初重徴	360	<code>\lc</code>
初重羽	320	<code>\lw</code>
二重宮	270	<code>\q</code>
二重商	230	<code>\s</code>
二重角	180	<code>\k</code>
二重徴	140	<code>\c</code>
二重羽	90	<code>\w</code>
三重宮	40	<code>\hq</code>
三重商	0	<code>\hs</code>
三重角	-45	<code>\hk</code>

表 2. 基本音符

音階名	命令	別名	音階名	命令	別名
初重徴	<code>\lchi</code>	初徴			
初重羽	<code>\lwoo</code>	初羽	揚初羽	<code>\yolwoo</code>	揚初羽
二重宮	<code>\qyu</code>	宮	反宮	<code>\hnqyu</code>	反宮
二重商	<code>\sho</code>	商	揚商	<code>\yosh</code>	揚商
二重角	<code>\kak</code>	角			
二重徴	<code>\chi</code>	徴	反徴	<code>\hnchi</code>	反徴
二重羽	<code>\woo</code>	羽	揚羽	<code>\yowoo</code>	揚羽
三重宮	<code>\hqyu</code>	三宮	反三宮	<code>\hnhqyu</code>	反三宮
三重商	<code>\hsho</code>	三商	揚三商	<code>\yohsho</code>	揚三商
三重角	<code>\hkak</code>	三角			

長さの規定の基本単位は英字サイズの半分。  
 連続した命令は連結した楽譜を出力。

例 `\tanni=1em`

ソース	出力
<code>\hakase{\宮}</code>	
<code>\hakase{\商}</code>	/
<code>\hakase{\宮\商\角\徴}</code>	└─┘
<code>\hakase{\羽\揚羽\三宮}</code>	└─┘

### 3.2 音の装飾を指定する音符

作図命令の基本パターンは

`\音符命令{音階}`

である。以下使用可能な命令を表にまとめる。音符命令欄が複数行になっている時は別名が定義されている。\*がついた名前は音符を区別するために便宜上命名した仮の名前である。

名前	音符命令	使用例	結果
律のユリ	<code>\yuriR</code> <code>\律由</code>	<code>\律由{\c}</code>	
呂のユリ	<code>\yuriL</code> <code>\呂由</code>	<code>\呂由{\c}</code>	
由下	<code>\yuge</code> <code>\由下</code>	<code>\由下{\c}</code>	
ツキユリ	<code>\tsukiyuri</code> <code>\ツキユリ</code>	<code>\ツキユリ{\c}</code>	
力のソリ	<code>\chikara</code> <code>\力</code>	<code>\力{\s}</code>	
ソリ	<code>\sori</code> <code>\ソリ</code>	<code>\ソリ{\s}</code>	
ソル	<code>\soru</code> <code>\ソル</code>	<code>\ソル{\s}</code>	
ソリハネ	<code>\sorihane</code> <code>\ソリハネ</code>	<code>\ソリハネ{\w}</code>	
ソリキリ	<code>\sorikiri</code> <code>\ソリキリ</code>	<code>\ソリキリ{\s}</code>	
ユリソリ	<code>\yurisori</code> <code>\ユリソリ</code> <code>\由ソ</code>	<code>\ユリソリ{\s}</code>	
荒由	<code>\arayu</code> <code>\荒由</code>	<code>\荒由{\c}</code>	
ユリカケ切り	<code>\yurikake</code> <code>\ユリカケ</code>	<code>\ユリカケ{\c}</code>	
イロ	<code>\iro</code> <code>\イロ</code>	<code>\イロ{\k}</code>	
イロトメ*	<code>\irotome</code> <code>\イロトメ</code>	<code>\イロトメ{\k}</code>	
イロ上*	<code>\iroage</code> <code>\イロ上</code>	<code>\イロ上{\k}</code>	
イロハネ*	<code>\irohane</code> <code>\イロハネ</code>	<code>\イロハネ{\k}</code>	
ハヌル	<code>\hanuru</code> <code>\ハヌル</code>	<code>\ハヌル{\s}</code>	
下ゲモドリ*	<code>\sagemodori</code> <code>\下ゲモドリ</code>	<code>\下ゲモドリ{\hq}</code>	
フリユリ	<code>\furiyuri</code> <code>\フリユリ</code>	<code>\フリユリ{\k}</code>	
四由	<code>\shiyu</code> <code>\四由</code>	<code>\四由{\k}</code>	
フル	<code>\furu</code> <code>\フル</code>	<code>\フル{\s}</code>	
ユリアゲ*	<code>\yuriage</code> <code>\ユリアゲ</code>	<code>\ユリアゲ{\c}</code>	
根音	<code>\negoe</code> <code>\根音</code>	<code>\根音{\c}</code>	

table continued on next page

continued from previous page

名前	音符命令	使用例	結果
引ク*	\hiku \引ク	\引ク{\s}	ノ
押ス	\osu \押ス	\押ス{\c}	ㇿ
大ユ	\ooyu \大ユ	\大ユ{\q}	ㇾ
ツヤ	\tsuya \ツヤ	\ツヤ{\s}	ノ
ツヤモチ*	\tsuyamochi \ツヤモチ	\ツヤモチ{\s}	ノ
ユルグ	\yurugu \ユルグ	\ユルグ{\s}	ノ
アタル	\ataru \アタル	\アタル{\k}	ㇿ
打付	\uchitsuke \打付	\打付{\s}	・
カカル	\kakaruru \カカル	\カカル{\k}	ㇿ
受音	\ukekoe \受音	\受音{\k}	ㇿ
折捨	\orisute \折捨	\woo\折捨{\k}	ノ

### 3.3 その他の音符

---

\modori\*    \modori\*( <音階方向> : <音符長> ) [ <連結部> ]  
\modori    \modori{ <相対音階> } [ <連結部> ]  
\戻        \modori{ <相対音階> }  
\モドリ    \modori = \modori{-135}, \modori{-145}: 商音のみ  
\modoli    \戻        = \modori の別名  
            \モドリ    = \modori の別名

---

音階方向: 音階を表す角度

相対音階: 現在の音階に対する相対角度

音符長    : 1\tanni の倍数

連結部    : 繋ぎ空白の伸びる方向

p        : 前音階方向に空白が伸長後戻り符を描画

n        : 前符終点で屈曲後モドリ音階方向に空白が伸長

他        : 前符とモドリの中間で折れ曲がる

連続する音符が干渉する場合のために戻り角度と始点の位置を変えた戻りも定義している。

\modoli = \modori{-110}[p]

例) \角\modori\角    ㇿ

---

\iromodori    \iromodori{ <音階方向> }  
\イロモドリ    \iromodori = \iromodori{-135}  
                 \イロモドリ = \iromodori の別名

---

例) \角\iromodori\角    ㇿ

---

`\kiri`      切り、息継ぎ  
`\きり`      音階方向に向かって右側にマーク  
`\kiri` [ <方位指定> ]  
`\kiri` = `\kiri` [ <現在の音階の角度>- 90]  
`\きり` = `\kiri` の別名  
方位指定方法は前述。  
例) `\chi\kiri\chi`

---



---

`\kili`      音階方向に向かって左側にマーク  
`\キリ`      `\kili` [ <方位指定> ]  
`\kili` = `\kiri` [ <現在の音階の角度> + 90]  
`\キリ` = `\kili` の別名  
例) `\chi\kili\chi`

---



---

`\tsu`      `\tsu`  
ツマル音  
例) `\chi\tsu`

---



---

`\mawasu`      マウス一音ずつ連続的に音を下げる  
`\マウス`      `\mawasu` [ <回数> ] { <初期音階> }  
`\マウス` = `\mawasu` の別名  
例) `\mawase[3]{\w}`

---



---

`\jige`      `\jige`  
`\自下`      `\自下`  
  
本自下     

---



---

`\nakaoroshi`      `\nakaoroshi` { <音階方向> }  
`\中下`      `\nakaoroshi` = `\nakaoroshi` { \k }  
`\中下` = `\nakaoroshi` の別名     

---



---

`\nagashi`      `\nagashi` { <音階方向> }  
`\流し`      `\nagashi` = `\nagashi` { \k }  
`\流し` = `\nagashi` の別名     

---



---

`\kakari`      `\kakari` { <音階方向> }  
`\カカリ`      `\kakari` = `\kakari` の別名     

---



---

`\kanaatari`      `\kanaatari` { <音階方向> }  
`\カナ当り`      `\kanaatari` = `\kanaatari` { \c }  
`\カナ当` = `\kanaatari` の別名  
`\カナ当り` = `\kanaatari` の別名  
`\kanaAtari` = `\kanaatari` { \nl{0.7} } \chi     

---

### 3.4 移動命令

---

`\moveTo`      `\moveTo` ( <x 方向変位>, <y 方向変位> ) : 現在位置からの相対移動。

---



---

`\kaigyo`      `\kaigyo` { <改行数> } : 次の描画開始位置を-y 方向に <改行数> \tanni ずらす。  
`\nl`      `\kaigyo` : 次の描画開始原点を-y 方向に 1\tanni ずらす。  
`\nl`      : 同上

---

---

**\aki**    **\aki**[ <変位> ]{ <進行方向> } : 現在の位置から <進行方向> へ <変位> \tanni 相対移動  
**\aki**{ <進行方向> } : 現在の位置から <進行方向> へ 0.5\tanni 相対移動  
**\aki** : 現在の位置から現在の描画進行方向 (音階方向) へ 0.5\tanni 相対移動  
 音譜と音符の間に空白を入れる。

### 3.5 文字出力

---

**\moji**    **\moji**( <相対 x>, <相対 y> ) [ <方位> ] { <文字列> }  
**\moji**[ <方位> ] { <文字列> } = **\moji**(0,0)[方位]{文字列}  
 相対座標は音階方向を x 軸正方向、それに垂直右方向を y 軸正方向とする。現在の描画位置から変位 (x,y) だけ移動した位置で指定した方位に文字を出力する。y 軸方向の文字描画開始位置はあらかじめずらしてあるので微調整のために使用されたし。

**方位**

方位は紙面上方を北とし、8 等分した方位をアルファベット又は数字で指定する。

東	北東	北	北西	西	南西	南	南東
e	ne	n	nw	w	sw	s	se
0	1	2	3	4	5	6	7

例) **\chi\chi\moji**(-0.3,0)[ne]{長}

長 ↘

---

**\chou**    長短 : 音の長さを指定  
**\長**    **\chou**[ <方位指定> ]  
**\tann**    **\長** = **\chou** の別名  
**\矢**    **\tann**[ <方位指定> ]  
**\矢** = **\tann** の別名、短の字の偏 (へん)

例) **\chi\矢** [sw]**\chi\長** [ne]

長 ↘  
矢

---

**\hikidashi**    音譜と文字列を線で繋ぐ  
**\hikidashi**( <x1>, <y1> )( <x2>, <y2> ) [ <方位> ] { <文字列> }  
**\hikidashi**( <x1>, <y1> )( <x2>, <y2> ) { <文字列> }  
**\hikidashi**( <x>, <y> ) [ <方位> ] { <文字列> }  
**\hikidashi**( <x>, <y> ) { <文字列> } = **\hikidashi**(x,0)(x,y)[s]{文字列}  
 現在の位置から相対座標 (x1,y1) を経由して (x2,y2) まで折線を引き、(x2,y2) の周囲の指定した方向を起点に文字列を出力する。方位の規定値は南である。