

luatexja-ruby パッケージ

LuaTeX-ja プロジェクトチーム

2020-10-05 v0.3 (2020 年 10 月 5 日)

概要

luatexja-ruby パッケージは, LuaTeX-ja の機能を利用してルビの組版処理を行う追加パッケージである. LuaTeX, LuaTeX-ja の内部処理に割り込むことにより, 熟語ルビ中の行分割や, 行頭形・行中形・行末形の自動検出, また進入許容量の自動設定などを可能とした.

v0.3 より前とは親文字の高さの扱いが変わっている (`baseheight` キーを参照せよ) ほか, `rubypreintrusion`, `rubypostintrusion` パラメータの初期値が変わっているので注意すること.

目次

1	<small>りようほうほう</small> 利用方法	2
1.1	<small>ようご</small> 用語	2
1.2	<small>めいれい</small> 命令	2
1.3	<small>してい</small> グループの指定	5
2	<small>ちゅういてん</small> 注意点	6
3	<small>じつそう</small> 実装について	7
4	いくつかの例	9
5	『日本語組版処理の要件』20120403 の例	11

1 利用方法

パッケージ読み込みは、`\usepackage{luatexja-ruby}` で良く、必要ならば自動的に LuaTeX-japan 本体を読み込む。plain LuaTeX でのロードはまだサポートしておらず、 $\text{\LaTeX} 2_{\epsilon}$ のみサポートしている。

1.1 用語

「進入 (intrusion)」「突出 (protrusion)」という用語は、`pxrubrica` パッケージでの用法に準ずる。

進入あり：あかつきと暁の
進入なし：あかつきと暁の
突出あり：ちようしゆう聴衆
突出なし：ちようしゆう聴衆

1.2 命令

■`\ltjruby` ルビ出力用命令の本体。 `\ruby` という別名を定義している。

`\ltjruby[<option>]{親|文|字}{おや|も|じ}`

のように使用する。第 2・第 3 引数内の「|」はグループの区切りを表す。詳細は 1.3 小節を参照。

<option> には以下の内容を key-value リストで指定可能である：

pre=*<real>* 前進入許容量をルビ全角単位で指定。負の長さを指定した場合は、ルビの状況や直前の文字に応じた自動指定を意味する。既定値は負（つまり、自動指定）。

post=*<real>* 同様に、後進入許容量を指定する。既定値は負（自動指定）。

mode 進入処理のモードを表す bit vector。下位 2 bit は、pre や post が負である場合にしか効力を発揮しない。既定値は $(0001)_2 = 1$ 。

bit 0 進入を無効にするならば 0、有効にするならば 1。

bit 1 前進入許容量 B と後進入許容量 A が異なった場合、そのまま処理する場合は 0、小さい方に揃えるならば 1。

bit 2-3 ルビ文字の突出量 x から実際の前進入量 b 、後進入量 a の計算方法を指定する。親文字の文字数が $k + 1$ 、親文字の前に入る空白量・間の空白量・後ろの空白量の比が $p : q : r$ のとき、

00 $b = \min(B, xp/(p + kq + r))$, $a = \min(A, xr/(p + kq + r))$

01 $b = \min(B, x)$, $a = \min(A, \max(x - b, 0))$

10 $a = \min(A, x)$, $b = \min(B, \max(x - a, 0))$

11 $M = \min(B, A)$ とおく。もし $x \leq 2M$ ならば $b = a = x/2$ 。そうでなければ

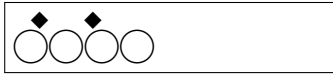
$$b = \min\left(B, M + \frac{(x - 2M)p}{p + kq + r}\right), \quad a = \min\left(A, M + \frac{(x - 2M)r}{p + kq + r}\right)$$

組み方の具体例を実際に示す。例示のため、平仮名にはルビが 1 字まで、「立」にはルビを 0.5 字分までかけてよいことにしている。

00 うつくは美しい ちようしゆうは聴衆と あかつきは暁立 ちようしゆうは聴衆立
01 うつくは美しい ちようしゆうは聴衆と あかつきは暁立 ちようしゆうは聴衆立
10 うつくは美しい ちようしゆうは聴衆と あかつきは暁立 ちようしゆうは聴衆立

11 ^{うつく}は美しい ^{ちようしゆう}は聴衆と ^{あかつき}は暁立 ^{ちようしゆう}は聴衆立

stretchruby={*left*}{*middle*}{*right*} 親文字の合計長がルビ文字の合計長より長い時に、ルビ文字の前・間・後に入れる空白の割合であり、それぞれ 0-7 の自然数で指定する。既定値は {1}{2}{1} である。$left$ はルビ文字の先頭までの空き量、$middle$ はルビ文字間の空き量、$right$ はルビ文字の末尾からの空き量 (の比) を表す。以下が例である。



```
1 \ltjruby[stretchruby=123,maxmargin=2]%
2 {○○○○}{◆◆}
```

stretch={*left*}{*middle*}{*right*} 行中形でルビ文字の方が長い場合、親文字の前・中・後に入れる空白の割合。既定値は {1}{2}{1} である。それ以外の代表的な値としては、例えば次のようなものがある。

親文字均等割禁止 {1}{0}{1} など $middle$ を 0 にした値

前突出禁止 {0}{1}{1}

後突出禁止 {1}{1}{0}

stretchbol={*left*}{*middle*}{*right*} 行頭形〜。既定値は {0}{1}{1} である。

stretcheol={*left*}{*middle*}{*right*} 行末形〜。既定値は {1}{1}{0} である。

maxmargin=$real$ 親文字の方がルビより長い時に、ルビの先頭と親文字の先頭、及びルビ末尾と親文字の末尾の間に許される最大の空白量。親文字全角単位で指定し、既定値は 0.5。

size=$real$ ルビ文字の親文字に対する大きさ。既定値は 0.5。

intergap=$real$ ルビ文字と親文字との空きを親文字全角単位で指定。既定値は 0。

rubymash=$bool$ ルビの高さを 0 にするか。既定値は偽。次が例である。

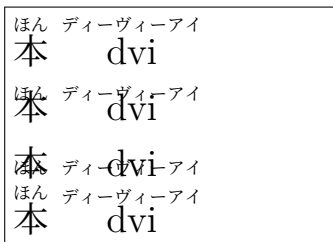


```
1 \vrule width 0pt height 2\zw depth 1\zw
2 \frame{\ltjruby[rubymash=false]{本}{ほん}}\
3 \frame{\ltjruby[rubymash=true]{本}{ほん}}\
4 \frame{\ltjruby[rubymash=false,intergap=0.2]
5 {本}{ほん}}\
6 \frame{\ltjruby[rubymash=true,intergap=0.2]
7 {本}{ほん}}\
8 \frame{\ltjruby[rubymash=false,intergap=-1.5]
9 {本}{ほん}}
```

ybaseheight=$real$ 非負の値が指定された場合、縦組以外でのルビの親文字の高さを全角高さの $real$ 倍と強制的に固定する。負の値が指定された場合は「固定しない」(すなわち、v0.3 より前の挙動と同じになる)。既定値は 0.88。

tbaseheight=$real$ ybaselineshift と同様だが、こちらは縦組でのルビの親文字の高さを指定する。既定値は 0.5。

baseheight=$real$ ybaselineshift, tbaselinesift を同時に指定したことと同義。



```
1 \noindent
2 \ltjruby[baseheight=0.88]{本}{ほん}\
3 \ltjruby[baseheight=0.88]{dvi}{ディーヴィーアイ}\
4 \ltjruby[baseheight=0.5]{本}{ほん}\
5 \ltjruby[baseheight=0.5]{dvi}{ディーヴィーアイ}\
6 \ltjruby[baseheight=0]{本}{ほん}\
7 \ltjruby[baseheight=0]{dvi}{ディーヴィーアイ}\
8 \ltjruby[baseheight=-1]{本}{ほん}\
9 \ltjruby[baseheight=-1]{dvi}{ディーヴィーアイ}
```

kenten=$command$ 各文字につく圏点の出力命令を指定する。既定値は「\ltjalchar`・」であり、U+2022 (Bullet) を ALchar として出力することを指定している。

以上で記述されていない `rubypreintrusion`, `rubypostintrusion` の初期値はすべて 0 である。

■`\ltjkenten` 圏点を出力する命令で、`\kenten` という別名を定義している。

`\ltjkenten[<option>]{親文字}`

のように使用する。複数文字に圏点をつける場合でも、`\ltjruby` のように「|」を使って文字を区切る必要はないことに注意してほしい。

圏点として出力される文字は `kenten` キーによって指定し、圏点自身の大きさは（ルビと同様に）`size` キーで指定する。

1.3 グループの指定^{してい}

`\ltjruby[<option>]{親|文|字}{おや|も|じ}`

のように、`\ltjruby` の第 2・第 3 引数内の「|」はグループの区切りを表す。グループの数は両者で一致しなければならず、`\ltjruby{紋章}{もん|しよう}` のようにはできない。

1 グループのみのルビ（単純グループルビ）はグループルビとして組まれる。そのため、もしモノルビを使いたければ、面倒でも

`もんしよう`
の紋章が

`\ltjruby{紋}{もん}\ltjruby{章}{しよう}`

のように、複数回使用すること。また、全てのグループにおいて「ルビ文字列の長さは親文字列以下」であれば、単純グループルビの並びとして扱われる。すなわち、次の 2 行は全くの等価となる。

`\ltjruby{普|通|車}{ふ|つう|しや}`

`\ltjruby{普}{ふ}\ltjruby{通}{つう}\ltjruby{車}{しや}`

複数グループかつ上記の条件を満たさない場合は、所謂「可動グループルビ」であり、ルビの前後や各グループの切れ目で行分割が可能となる。例えば

…の`\ltjruby{表|現|力}{ひよう|げん|りよく}`は…

という入力からは得られる組版結果は、次のいずれかになる。

改行なし（行中形）	ひようげんりよく …の表現力は…
直前で改行	ひようげんりよく 表現力は…
	ひようげんりよく …の表現力は…
	ひようげんりよく …の表現力は…
直後に改行	ひようげんりよく …の表現力

- これらの行分割によってペナルティは発生しない。
- 上記の例で見られるように、2 ブロック以上をまとめて組むときは、全体を 1 つのグループルビのように組版する（JIS X 4051 と同様）。『日本語組版処理の要件』では、附属書 F に「熟語の構成、さらにその熟語の前後にくる文字の種類を考慮して配置する方法」として別の方法を解説しているが、こちらの方法は現時点ではサポートしていないので、面倒でも手動で `pre`, `post` などを使って頑張りたい。
- 実装方法の都合により、ルビの直前・直後・途中で 2 箇所以上の改行が起きる場合、例えば

などの組み方は想定していない。

2 注意^{ちゆういてん}点

ルビ文字のはみ出し 「日本語組版処理の要件」の図 3.82 のように、ルビ文字のはみ出しが繋がらないようにする処理について述べる。

例えば、

^{りよう} ^{みささぎ} 陵と陵 ^{りよう} ^{みささぎ} 陵と陵
--

- 1 `\ltjruby{陵}{りよう}`と`\ltjruby{陵}{みささぎ}\`
- 2 `\ltjruby{陵}{りよう}`と`\ltjruby[pre=1]{陵}{みささぎ}`

において、1 行目右側の「陵」のルビが前の「と」にかかる量は次のように決まる：

1. 1 回目の実行では、行分割前に「陵」の後側進入量は前もって知ることはできない。なので、「陵」は行中形で組まれるものとして「陵」前側進入許容量は

$$\frac{0.5zw}{\text{元々の許容量}} - \frac{0.25zw}{\text{前のルビの後側進入量 (行中形)}} = 0.25zw$$

となる。行分割後、「陵」の実際の後側進入量は `<jobname>.ltjruby` ファイルに記述される。

2. 2 回目以降の実行では、`<jobname>.ltjruby` ファイルに保存された「陵」の後側進入量を用いて、「陵」前側進入許容量を次のように計算する：

$$\frac{0.5zw}{\text{元々の許容量}} - \frac{0.25zw}{\text{前のルビの後側進入量 (from .ltjruby)}} = 0.25zw.$$

`<jobname>.ltjruby` ファイルに保存する際、各 `\ltjruby` 命令の呼び出しを識別するキーが必要になるが、そのキーとしては単純に「何個目の `\ltjruby` 命令か」である。

なお、以上の処理は、1 行目と 2 行目を比較すれば分かるように、「陵」の前側進入許容量指定 (`pre`) が自動になっている場合のみ実施される。

段落末尾のルビ 段落がルビで終わった場合、そのルビが行末形で組まれることはない。これは、段落の「本当の」末尾には `\penalty10000\parfillskip` があるためで、ルビ処理用に作った最後のグルー (3 節の説明では g_2) が消去されないことによる。

`\parfillskip` の長さ (や、場合によっては `\rightskip`) を実測し、それによって処理を変えるのも可能だが、そのようなことはしなかった。段落がルビで終わることは普通ない (最低でも句点が続くだろう) と思うからである。

段落先頭のルビ 同様に、段落先頭のルビは行頭形にはならない。 `pre` が負 (つまり、自動指定) のとき、段落最初の `\parindent` 部分への進入は可能である。 `\parindent` 部分への進入許容量は「文字 -1」に対する `rubypreintrusion` (既定値は 1, ルビ全角単位) と `\parindent` の長さのうち小さい方である。

0 1 2 3 4 5 ^{みささぎ} 陵は…… ^{うけたまわ} 承り…… ^{みささぎ} 陵は……

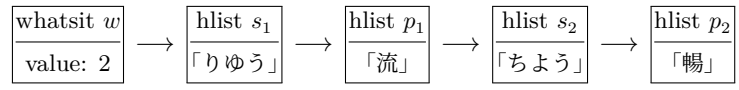
- 1 `\parindent1\zw\noindent 0 1 2 3 4 5\par`
- 2 `\ltjruby{陵}{みささぎ}は……\par`
- 3 `\ltjruby{承}{うけたまわ}り……\par`
- 4 `\parindent0.25\zw\ltjruby{陵}{みささぎ}は……`

3 実装について

次の例で実装の大きな方法を説明する。

$\dots\dots$ <small>りゅうちよう</small> を流暢に $\dots\dots$	<code>1 \dots\dotsを\ltjruby{流 暢}{りゅう ちよう}に\dots\dots</code>
--	---

1. `\ltjruby` コマンド自体は、一旦次の node list を値とする whatsit W を作って、現在の水平リストへと挿入する（必要ならば `\leavevmode` も実行）：

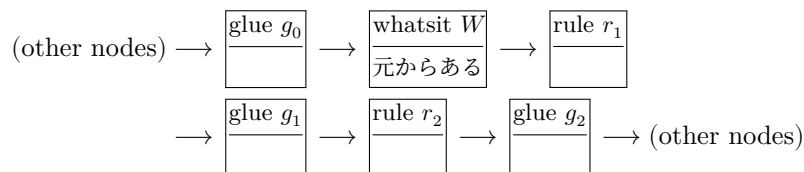


ここで、最初の w の値 2 は、ルビが 2 つのパーツ「流」「暢」からなっていることを表している。この値を cmp とおこう。 s_i 達の中の文字は既にルビの大きさである。

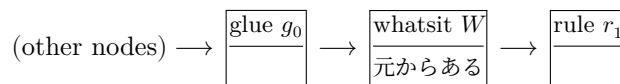
2. LuaTeX-ja の和文処理グルー挿入処理において、この whatsit W はまとめて「先頭が『流』、最後が『暢』であるような hbox を `\unhbox` で展開したもの」と扱われる。言い換えれば、ルビ部分を無視した単なる「流暢」という和文文字の並びとして扱われる*2。次のサンプルを参照

\cdot A \cdot A	<code>1 \leavevmode\hbox{. }A\</code> <code>2 %\upkanjiskip</code> <code>3 \ltjruby{. }{A}</code> <code>4 %\up2分</code>
------------------------	--

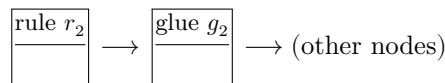
3. 和文処理グルーの挿入が終わった後で、可動グルーのルビのためのノードの挿入に入る。
 (a) W の前後に $2cmp + 1 = 5$ 個のノードが挿入され、 W の周辺は次のようなノード列になる。



- (b) このようにノードを挿入する目的は、TeX の行分割処理自体に影響を加えずに可動グルーのルビを実現させることにある。



行の境目



のようになったとしたら、「流」「暢」の間で行分割が起きた、ということがわかり、 g_i, r_i 達のノードを適切に置き換えればよい（後で詳しく説明する）。

- (c) なお、 r_i 達の高さ・深さは組み上がった後のそれである。 g_i, r_i 達の幅は、図 1 に示したような対応に沿って算出する。例えばこの場合、行中形 n_5 に対して

$$g_0 + r_1 + g_2 + r_2 + g_2 = 3zw - (0.25zw \times 2) = 2.5zw$$

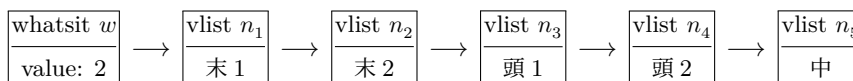
という方程式が立つ (zw は親文字全角の幅、進入量込)。 n_1 から n_5 まで計 5 本の方程式が立つが、これらは Gauß の消去法で解くことができ、 g_i, r_i 達の幅が求まる。

*2 「流」「暢」の間のグルーは既に入っている、と扱われる。

ノード	組み方	サンプル	対応するノード並び
n_1	行末 1 グループ	を流	$g_0 \rightarrow W \rightarrow r_1$
n_2	行末 2 グループ	を流暢	$g_0 \rightarrow W \rightarrow r_1 \rightarrow g_2 \rightarrow r_2$
n_3	行頭 1 グループ	暢に	$r_2 \rightarrow g_2$
n_4	行頭 2 グループ	流暢に	$W \rightarrow r_1 \rightarrow g_2 \rightarrow r_2 \rightarrow g_2$
n_5	行中	を流暢に	$g_0 \rightarrow W \rightarrow r_1 \rightarrow g_2 \rightarrow r_2 \rightarrow g_2$

図 1 ルビの組み方と対応するノード並び

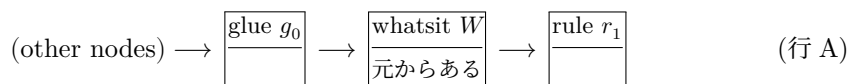
(d) また、ルビ処理を統括している whatsit W の値も



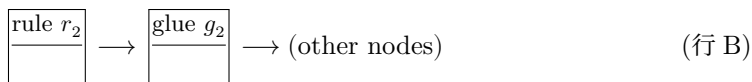
に置き換えておく。

4. LuaTeX の行分割処理を普通に行う。
5. 行分割の結果に従って、 g_i, r_i 達を適切に置換する。

例えば行分割の結果



行の境目



のようになったとしよう。

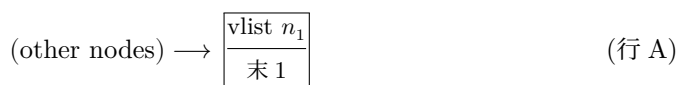
- (a) 処理は段落の上の行から順番に行われる。行 A の処理がまわってきたとしよう。
- (b) 行 A の先頭から順番に眺めていく。すると「whatsit W 由来」のノード、 g_0, W, r_1 が見つかり、行 A はここで終わっている。

まず、行 A の hbox の中身から whatsit W を消去（リストから取り除くだけで、 W のメモリを解放するわけではない）する。 $g_0, (W,) r_1$ というノードの並びは、「行末 1 グループ」 n_1 に対応しているので、 g_0, r_1 を行 A から除去・メモリ解放し、代わりに n_1 を行 A の中身に追加する。

- (c) 次に行 B の処理にうつる。行 A でルビの処理は完了していない（2 グループのルビなのにまだ 1 グループしか使っていない）ので、「whatsit W 由来」のノードがいくつか行 B 内に残っているはずである。

案の定、 r_2, g_2 というノード列が見つかった。これは「行頭 1 グループ」 n_3 に対応しているので、 r_2, g_2 を行 B から除去・メモリ解放し、代わりに n_3 を行 B の中身に挿入する。

- (d) これで 2 グループとも使い切ったことになるので、隔離しておいた W を、（使われなかった n_2, n_4, n_5 などと共に）メモリ解放する。結果として次のようになった：



行の境目



4 いくつかの例

ゴールデンゲートブリッジ
 あああ黄金橋いうえおかきくけこあ
ゴールデンゲートブリッジ ゴールデンゲートブリッジ
 黄金橋いうえおかきくけこあ黄金橋
ゴールデンゲートブリッジ
 いうえおかきくけこあ黄金橋いうえおか
ゴールデンゲートブリッジ
 きくけこあ黄金橋いうえおかきくけこあ
ゴールデンゲートブリッジ
 黄金橋いうえおかきくけこ

baseheight=0.88での例
ディーヴィーアイ ほーげーふーが
 ふぁいる dvi ファイル oo 漢字
ディーヴィーアイ ほーげーふーが
 ファイル dvi ふぁいる oo 漢字
ふがふがふが ふがふがふが ふがふがふが ふがふがふが
 ああ(ほげ) ほげ 【ほげ】 ほげ...

こうづ
 あ国府津いうえおかきくけこあ国府津
こうづ
 いうえおかきくけこあ国府津いうえおか
こうづ
 きくけこあ国府津いうえおかきくけこあ
こうづ
 国府津いうえおかきくけこあ国府津いう
こうづ
 えおかきくけこ

● ◆
 あ●◆いうえおかきくけこあ●◆
● ◆
 ◆いうえおかきくけこあ●◆◆いうえお
● ◆
 かきくけこあ●◆◆いうえおかきくけこ
● ◆
 イあ●◆◆いうえおかきくけこあ●◆◆
● ◆
 ◆いうえおかきくけこウあ●◆◆いう
● ◆
 えおかきくけこエあ●◆◆いうえおか
● ◆
 きくけこあ●◆◆いうえおかきくけこあ
● ◆
 ●◆◆いうえおかきくけこ

● □
 あ●□いうえおかきくけこあ●□い
● □
 うえおかきくけこあ●□いうえおかきく
● □
 けこあ●□□いうえおかきくけこイあ●
● □
 □いうえおかきくけこあ●□□いうえお
● □
 かきくけこウあ●□□いうえおかきくけこ
● □
 エあ●□□いうえおかきくけこあ●□□い
● □
 うえおかきくけこあ●□□いうえおかきく
● □
 けこ

いよう
 あ異様いうえくけあ異様いうえくけこあ
いよう
 異様いうえくけこあ異様いうえくけこイあ
いよう
 異様いうえおかきくけこ

■ □
 あ■□□いうえおかきくけこうえおかきくけこ
■ □
 あ■□□いうえおかきくけこうえおかきくけこあ
■ □
 □■□□いうおかきくけこうえおかきくけこあ
■ □
 ■□□□いう□おかきくけこうえおかきくけこあ
■ □
 □いう□おかきくけこうえおかきくけこあ□
■ □
 □いう□おかきくけこうえおかきくけこあ□
■ □
 いうえおかきくけこ

うけたまわ そ にわか にわか あかつき にわか きゆうけいちゆう しちようちゆう
 標準 又 承る 疎 は俄勉強 後 俄勉強 は暁には俄に 休憩 中 かつ 視聴 中
うけたまわ そ にわか にわか あかつき にわか きゆうけいちゆう しちようちゆう
 肩つき 又 承る 疎 は俄勉強 後 俄勉強 は暁には俄に 休憩 中 かつ 視聴 中

<small>きゆうけいちゆう</small> 休憩中	<small>しちようちゆう</small> 視聴中	<small>べんきよう</small> <small>べんきよう</small> 勉強と勉強	<small>Cogito ergo sum</small> コギト・エルゴ・スム	<small>Cogito ergo sum</small> コギト・エルゴ・スム
--------------------------------	-------------------------------	--	--	--

```

1 {\ltjsetruby{stretch=101}% 親文字均等割り禁止
2   \ruby{休|中}{きゆう|けい|ちゆう}かつ\ruby{視|聴|中}{し|ちよう|ちゆう}%
3 }\quad
4 \textgt{\ruby{勉|強}{べん|きよう}と}%
5   \ruby[fontcmd=\mcfamily]{勉|強}{べん|きよう}\quad% ルビは明朝体
6 \ruby{コギト・エルゴ・スム}{Cogito ergo sum}\quad % 欧文空白は伸長しない
7 \ruby[size=1]{Cogito ergo sum}{コギト・エルゴ・スム}% 欧文空白は伸長しない

```

5 『日本語組版処理の要件』 20120403 の例

■3.3.1 節

3.49	<small>くんし わ どう</small> 君子は和して同ぜず
3.50	<small>ひと おし う</small> 人に誨えて倦まず
3.51	<small>きもん ほうがく ぎょうし</small> 鬼門の方角を凝視する
3.52	<small>きもん ほうがく ぎょうし</small> 鬼門の方角を凝視する
3.53	<small>かやばちよう かやばちよう</small> 茅場町 茅場町
3.54	<small>あじさい るつぼ いなか</small> 紫陽花 埧埧 田舎
3.55	<small>モデル クライアント インターフェース アジール</small> 模型 顧客 境界面 避難所
3.56	<small>editor エディター</small> 編集者 editor

```
1 \obeylines
2 3.49 \ruby{君子}{くん|し}は\ruby{和}{わ}して\ruby{同}{どう}ぜず
3 3.50 \ruby{人}{ひと}に\ruby{誨}{おし}えて\ruby{倦}{う}まず
4 % モノルビ．面倒でも複数回の実行が必要
5 3.51 \ruby{鬼}{き}\ruby{門}{もん}の\ruby{方}{ほう}\ruby{角}{がく}を%
6     \ruby{凝}{ぎょう}\ruby{視}{し}する
7 % 熟語ルビ
8 3.52 \ruby{鬼|門}{き|もん}の\ruby{方|角}{ほう|がく}を\ruby{凝|視}{ぎょう|し}する
9 3.53 \ruby{茅場町}{かやばちよう}\quad\ruby{茅|場}{かや|ば}\ruby{町}{ちよう}
10 % 熟字訓
11 3.54 \ruby{紫陽花}{あじさい}\quad\ruby{埧埧}{るつぼ}\quad\ruby{田舎}{いなか}
12 % グループルビ
13 3.55 \ruby{模型}{モデル}\quad\ruby{顧客}{クライアント}\quad%
14     \ruby{境界面}{インターフェース}\quad\ruby{避難所}{アジール}
15 3.56 \ruby{編集者}{editor}\quad \ruby{editor}{エディター}
```

■3.3.3 節

3.58	<small>のぼり</small> に幟を	<small>のぼり</small> に幟を	<small>いへんさんぜつ</small> 韋編三絶
3.59	<small>のぼり</small> に幟を		
3.60	<small>いへんさんぜつ</small> 韋編三絶	<small>いへんさんぜつ</small> 韋編三絶	

```
1 \obeylines
2 3.58 に\ruby{幟}{のぼり}を\quad に\ruby[kata]{幟}{のぼり}を\quad%
3     \ruby{韋}{い}\ruby{編}{へん}\ruby{三}{さん}\ruby{絶}{ぜつ}
4 % 三分ルビ．JY3/mc/mc は本文書のプリアンブルで独自に定義
5 3.59 に\ruby[fontcmd=\kanjifamily{mc}\kanjiserries{mc}]{幟}{のぼり}を
6 % ルビ文字を小さくする
7 3.60 {\Large%
8     \ruby{韋}{い}\ruby{編}{へん}\ruby{三}{さん}\ruby{絶}{ぜつ}\quad% 比較用
9     \ltjsetruby{size=0.375}% 0.5 -> 0.375
10    \ruby{韋}{い}\ruby{編}{へん}\ruby{三}{さん}\ruby{絶}{ぜつ}}
```

■3.3.4 節 3.61 図（両側ルビ）は未サポートにより省略

■3.3.5 節 モノルビ

3.62 の^{やく}約に
3.63 版面^ちの地^ちに 版面^ちの地^ちに
3.65 の^{とりで}砦に
3.66 上^{しゆん}の^{しゆん}句^{しゆん}に 後^{しゆん}句^{しゆん}に
3.66 下^{しゆん}の^{しゆん}句^{しゆん}又 後^{しゆん}句^{しゆん}又

```
1 \obeylines
2 3.62 の\ruby{約}{やく}に
3 3.63 版面の\ruby{地}{ち}に\quad 版面の\ruby[kata]{地}{ち}に
4 % 横組肩つきはしないが、現状では縦組未サポートだし、仕方ないね
5 3.65 の\ruby{砦}{とりで}に
6 {\ltjsetruby{kata}%
7 3.66上 の\ruby{句}{しゆん}に\quad 後\ruby{句}{しゆん}に
8 3.66下 の\ruby{句}{しゆん}又\quad 後\ruby{句}{しゆん}又}
```

■3.3.6 節 グループルビ

3.67 は^{コードツクス}冊子体と
3.68 ^{モデル}模型^{ライセンス} 利用許諾
3.69 ^{モデル}模型^{ライセンス} 利用許諾
3.70 ^{ピエタ}なげきの^エ聖母像^タ ^{ピエタ}なげきの^エ聖母像^タ
3.71 ^{クライアント}顧客^{インターフェース} 境界面
3.72 ^{クライアント}顧客^{インターフェース} 境界面

```
1 \obeylines
2 3.67 は\ruby{冊子体}{コードツクス}と
3 3.68 \ruby{模型}{モデル}\quad \ruby{利用許諾}{ライセンス}
4 % 両端を揃える流儀
5 3.69 {\ltjsetruby{stretchruby=010}%
6 \ruby{模型}{モデル}\quad \ruby{利用許諾}{ライセンス}}
7 % ルビが極端に短い場合
8 3.70 \ruby{なげきの聖母像}{ピエタ}\quad% ルビ全角まで許容
9 \ruby[maxmargin=0.75]{なげきの聖母像}{ピエタ}\% ルビ全角1.5倍まで
10 % ルビが長い場合
11 3.71 \ruby{顧客}{クライアント}\quad \ruby{境界面}{インターフェース}
12 3.72 {\ltjsetruby{stretch=010, stretchbol=010, stretcheol=010}% はみ出さない流儀
13 \ruby{顧客}{クライアント}\quad \ruby{境界面}{インターフェース}}
```

■3.3.7 節 熟語ルビ

3.73	^{きゆう} 杞憂	^{いふ} 畏怖	^{きゆう} 杞憂	^{いふ} 畏怖	
3.74	^{りゆうぎ} の流儀を	^{むじよう} の無常を	^{じようじゆ} の成就を	^{もんしよう} の紋章を	^{しyouちyou} の象徴を
3.75	^{りゆうぎ} の流儀を	^{むじよう} の無常を	^{じようじゆ} の成就を	^{もんしよう} の紋章を	^{しyouちyou} の象徴を
3.76×	^{りゆうぎ} の流儀を	^{むじよう} の無常を			
要調整 3.77	あああああああの			……等の無	
	^{りゆうぎ} 流儀がある。			←なぜ改行できない	
				^{じよう} 常を	

```

1 \obeylines
2 3.73 \ruby{杞|憂}{き|ゆう}\quad \ruby{畏|怖}{い|ふ}\quad%
3     \ruby[kata]{杞|憂}{き|ゆう}\quad \ruby[kata]{畏|怖}{い|ふ}
4 3.74 の\ruby{流|儀}{りゆう|ぎ}を\quad   の\ruby{無|常}{む|じよう}を\quad%
5     の\ruby{成|就}{じよう|じゆ}を\quad   の\ruby{紋|章}{もん|しyou}を\quad%
6     の\ruby{象|徴}{しyou|ちyou}を
7 % 熟語の構成を考慮した方法は現行ではできない.
8 % 行分割はできるのだが, 「他の漢字にルビ全角までかかって良い」は難しい.
9 3.75 {\ltjsetruby{kata}%
10     の\ruby{流|儀}{りゆう|ぎ}を\quad   の\ruby{無|常}{む|じよう}を\quad%
11     の\ruby{成|就}{じよう|じゆ}を\quad   の\ruby{紋|章}{もん|しyou}を\quad%
12     の\ruby{象|徴}{しyou|ちyou}を}
13 % モノルビ配置. 望ましくない
14 3.76× の\ruby{流}{りゆう}\ruby{儀}{ぎ}を\quad   の\ruby{無}{む}\ruby{常}{じyou}を\quad%
15
16 要調整 3.77 \{\ltjsetruby{stretchbol=121, stretcheol=121}% 行頭・行末揃えず
17     \hbox{\vrule\ vbox{\hspace=10\zw あああああああの%
18         \ruby{流|儀}{りゆう|ぎ}がある. }\vrule}\textbf{←なぜ改行できない}\quad%
19     \hbox{\vrule\ vbox{\hspace=5\zw ……等の\ruby{無|常}{む|じyou}を}\vrule}}

```

■3.3.8 節 ルビはみ出し

3.78	^{ひと} 人は死して	^な 名を	^{のこ} 残す
3.79	^{へん} 漢字の部首には偏・ ^{かんむり} 冠・ ^{きやく} 脚・ ^{つくり} 旁がある		
3.79	^{へん} 漢字の部首には偏, ^{かんむり} 冠, ^{きやく} 脚, ^{つくり} 旁がある		

```

1 \obeylines
2 3.78 \ruby{人}{ひと}は\ruby{死}{し}して\ruby{名}{な}を\ruby{残}{のこ}す
3 % 現在の自動設定の仕様では「脚」の前が四分だけ「要件」の例より空く
4 3.79 漢字の部首には\ruby{偏}{へん}・\ruby{冠}{かんむり}・\ruby{脚}{きやく}・%
5     \ruby{旁}{つくり}がある
6 3.79 漢字の部首には\ruby{偏}{へん}, \ruby{冠}{かんむり}, \ruby{脚}{きやく}, %
7     \ruby{旁}{つくり}がある

```

3.79	漢字の部首には偏・冠・脚・旁がある
3.79	漢字の部首には偏, 冠, 脚, 旁がある
3.79	この噂の好きな人は懐ぐあいもよく、檜を
3.80	漢字の部首には「偏」「冠」「脚」「旁」がある
3.80	この噂好きな人は懐 具合もよく、檜材を
3.81×	に暁の趣を (良くない例)
3.82	に暁の 趣を
3.83	この噂の好きな人は懐ぐあいもよく、檜を
3.83	この噂好きな人は懐 具合もよく、檜材を
3.84	この噂の好きな人は 懐 ぐあいもよく、檜を
3.84	この噂好きな人は 懐 具合もよく、檜材を

```

1 \obeylines
2 % もちろん手動で進入量を指定すれば話は別
3 3.79 漢字の部首には\ruby{偏}{へん}・\ruby{冠}{かんむり}・\ruby[pre=1]{脚}{きやく}・%
4   \ruby{旁}{つくり}がある
5 3.79 漢字の部首には\ruby{偏}{へん}, \ruby{冠}{かんむり}, \ruby[pre=1]{脚}{きやく}, %
6   \ruby{旁}{つくり}がある
7 3.79 この\ruby{\噂}{うわさ}の好きな人は\ruby{懐}{ふところ}ぐあいもよく、\ruby{檜}{ひのき}を
8 3.80 漢字の部首には「\ruby{偏}{へん}」「\ruby{冠}{かんむり}」「\ruby{脚}{きやく}」%
9   「\ruby{旁}{つくり}」がある
10 3.80 この\ruby{\噂}{うわさ}好きな人は\ruby{懐}{ふところ}具合もよく、\ruby{檜}{ひのき}材を
11 3.81× に\ruby{暁}{あかつき}の\ruby{趣}{おもむき}を (良くない例)
12 3.82 に\ruby{暁}{あかつき}の\ruby{趣}{おもむき}を
13
14 % 漢字にルビを2分まで掛けても良い流儀
15 {%
16   \tjsetparameter{rubypostintrusion}={`好,0.5}}
17   \tjsetparameter{rubypostintrusion}={`具,0.5}}
18   \tjsetparameter{rubypostintrusion}={`材,0.5}}
19 3.83 この\ruby{\噂}{うわさ}の好きな人は\ruby{懐}{ふところ}ぐあいもよく、\ruby{檜}{ひのき}を
20 3.83 この\ruby{\噂}{うわさ}好きな人は\ruby{懐}{ふところ}具合もよく、\ruby{檜}{ひのき}材を
21 }
22 % 平仮名にもルビを掛けない流儀
23 {\catcode`\<12%
24   \makeatletter\count@="3040\loop\relax\ifnum \count@<"30A0%
25   \tjsetparameter{rubypreintrusion}={\the\count@,0}, %
26     rubypostintrusion={\the\count@,0}}%
27   \advance\count@1 \repeat
28 3.84 この\ruby{\噂}{うわさ}の好きな人は\ruby{懐}{ふところ}ぐあいもよく、\ruby{檜}{ひのき}を
29 3.84 この\ruby{\噂}{うわさ}好きな人は\ruby{懐}{ふところ}具合もよく、\ruby{檜}{ひのき}材を
30 }

```

		……の徑 <small>こみち</small>
3.85	を…… 徑 <small>こみち</small> を……	……の ……の徑 <small>こみち</small>
3.86	を…… 徑 <small>こみち</small> を……	……の ……の
3.87	等…… 飾り等…… <small>アクセサリー</small>	……共 <small>アクセサリー</small> 飾り ……共

```

1 \obeylines
2 3.85\ \ltjsetruby{stretchbol=121, stretcheol=121}% 行頭・行末揃えず
3 \hbox{\vrule\vbox{\hsize=15\zw
4 \hskip9.5\zw……の\ruby{徑}{こみち}を……%
5 \hskip9\zw……の\ruby{徑}{こみち}を……}\vrule}
6 3.86\ \hbox{\vrule\vbox{\hsize=15\zw
7 \hskip10\zw……の\ruby{徑}{こみち}を……%
8 \hskip9\zw……の\ruby{徑}{こみち}を……}\vrule}
9 3.87\ \hbox{\vrule\vbox{\hsize=15\zw
10 \hskip8\zw……共\ruby{飾り}{アクセサリー}等……%
11 \hskip9\zw……共\ruby{飾り}{アクセサリー}等……}\vrule}

```

■**圈点の例**（常用漢字表前書きより） この表は、法令、公用文書、新聞、雑誌、放送など、一般の社会生活において現代の国語を書き表す場合の漢字使用の自安を示すものである。

「\」の大きさを親文字の0.33倍にした例 この表は、法令、公用文書、新聞、雑誌、放送など、一般の社会生活において現代の国語を書き表す場合の漢字使用の自安を示すものである。