

jlreq

1 これは何？

日本語組版処理の要件の実装を試みる [LuaTeX-ja](#) / pLaTeX / upLaTeX 用のクラスファイルと、それに必要な JFM の組み合わせです。

2 提供されるもの

クラスファイル `jlreq.cls` と、横書き LuaTeX-ja 用の JFM である `jfm-jlreq.lua` が用意されています。また、縦書きの JFM や pLaTeX / upLaTeX 用の JFM を生成するいくつかのスクリプトがあります。

3 インストール

`make` で必要な JFM を生成してください。その後、

- *.tfm -> \$TEXMF/fonts/tfm/public/jlreq
- *.vf -> \$TEXMF/fonts/vf/public/jlreq
- jfm-jlreq.lua, jfm-jlreqv.lua -> \$TEXMF/tex/luatex/jlreq
- jlreq.cls, jlreq-helpers.sty -> \$TEXMF/tex/latex/jlreq

と配置します。 `make install` とすると、`$TEXMF=$TEXMFHOME` としてこのコピーを行います。

4 使い方

通常通り

```
\documentclass{jlreq}
```

とします。これで横書きの `article` 相当の文書クラスとなります。エンジンは自動判定されますが、指定する場合はクラスオプションに `platex/uplatex/lualatex` のいずれかを渡してください。縦書きにするには `tate` オプションを渡します。また、`report` や `book` 相当の文書クラスとするには、それぞれ `report` や `book` オプションを渡します。たとえば、縦書きの本を作成するには

```
\documentclass[tate,book]{jlreq}
```

とします。

その他、 `oneside` / `twoside` / `onecolumn` / `twocolumn` / `titlepage` / `notitlepage` /

`draft / final / openright / openany / leqno / fleqn` というよくあるオプションを受け付けます。

標準的な文書クラスと同じように中身を書くことができますが、次のような命令が追加/ 拡張されています。なお、本ドキュメントでは[日本語組版処理の要件](#)における用語を断りなく使います。

4.1 `\jlrqsetup`

設定用命令です。プリアンブルでしか使えません。文書に対する設定は、クラスオプションとして行うか `\jlrqsetup` を通じて行うかします。どちらで設定するかは設定項目によります。

4.2 `\section`

`\section*[running head]{見出し文字列}[副題]` というように、通常の手式に加えて副題を受け付けられるように拡張されています。その他、`\part` (article のみ)、`\chapter` (book/report のみ)、`\subsection`、`\subsubsection` も副題を受け付けます。

4.3 abstract 環境

プリアンブルにもかけるようになっています。プリアンブルに書かれた場合は、`\maketitle` とともに出力されます。二段組の場合は、段組にならず概要を出力することができます。

4.4 `\sidenote`

この命令は傍注の幅が正の時にのみ定義されます。デフォルトの基本版面ではこの幅は0に設定されています。従って `\sidenote` は定義されません。後の基本版面の設定を参考にしてください。`\sidenote` は傍注（縦組みの場合は脚注）を出力します。内部では `\marginpar` を使っています。デフォルトでは `\footnote` と同様の書式となりますが、`\jlrqsetup` で `sidenote_type=symbol` が指定されている場合、その書式は `\sidenote{該当項目}{注}` となります。たとえば

刊行できる `\sidenote{原稿}{印刷などの方法により……}` を入手する仕事である。

とします。後の説明も参照してください。

4.5 `\endnote`

後注を指定します。`\footnote` と同様の書式です。デフォルトでは、注自身の出力は見出し直前に行われます。この動作は `\jlrqsetup` に `endnote_position` を渡すことで制御できます。詳しくは後の注関係の説明をご覧ください。また `\theendnotes` を実行するとその場に出力をします。

4.6 `\warichu`

割注を出力します。行分割位置などは自動で計算されます。（複数回のコンパイルが必要。）`\warichu*` ではこれらの位置を手動で指定できます。書式は

```
\warichu*{(一行目前) & (一行目後)}\ (二行目前) & (二行目後)...
```

です。&が省略されている場合は自動で調整されます。

4.7 \tatechuyoko

縦中横を出力します。 \tatechuyoko{<文字列>}とします。縦書きでない場所で使うとエラーになります。

4.8 \jafontsize

和文フォントサイズを指定する \fontsize です。クラスオプションで jafontscale=0.9 とされている場合、 \fontsize{9pt}{15pt}とすると和文フォントのサイズは 8.1pt となりますが、 \jafontsize{9pt}{15pt}とすると 9pt となります。(欧文フォントサイズは 10pt となる。) なお、第二引数は \fontsize の第二引数と全く同じです。

4.9 \

全角空白 (U+3000) 一文字からなるマクロです。和字間隔を挿入します。LuaLaTeX では のみでも和字間隔を入力できます。

4.10 その他

- ルビや圏点は提供されません。 [PXrubrica](#) または `luatexja-ruby` (LuaLaTeX, LuaTeX-ja パッケージに付属) を使うと良いかと思います。
- 日本語組版処理の要件 2.3.2.d によれば、横組みにおける二段組の最後のページの各段の行数は揃えることが望ましいとされていますが、この処理は行われません。 `nidanfloat` パッケージを使い、

```
\usepackage[balance]{nidanfloat}
```

とするとこの処理が行われます。ただし、最終ページでの \newpage や \clearpage が正しく動作しません。詳しくは `nidanfloat` パッケージのマニュアルをご覧ください。

- フォントを設定する機能は有していません。和文フォントは、LuaLaTeX 利用時は `luatexja-fontspec` や `luatexja-preset` (いずれも LuaTeX-ja パッケージに付属) により設定することができます。 `dvipdfmx` を使う場合は、 [PXchfon](#) での設定が可能です。

5 各種設計

設計はクラスオプションまたは \jlrqsetup により keyval 形式で行います。ただし、クラスオプションでは LaTeX の実装により、本来可能な入力が受け付けられないケースがあります。多くの

場合は空白を除くことにより解決します。

以下では次の用法を使います。

- [A/B] : A または B です。 [A/B/C] 等も同様。
- <寸法> : TeX が認識する寸法です。簡単な式 (10pt+10pt のような) を使うこともできます。また、クラスオプションでは、場合によっては次のような特殊な値を使うこともできます。(これらは pLaTeX / upLaTeX ではもともとから利用可能ですが、LuaLaTeX でも利用可能なように処理されています。) `\jlrreqsetup` 内のような場所では、常に `\zw` や `\zh` により全角幅が記述できます。以下、たとえば Q が利用可能な場合は <寸法;Q> のように記述します。
 - Q : 0.25mm と解釈されます。
 - zw, zh : 全角幅として解釈されます。
- <コード> : LaTeX のコードです。
- <フォント設定コード> : `\Large` や `\bfseries` のようなフォント設定の命令です。`\Large\bfseries` のように複数指定することもできます。

5.1 基本版面

クラスオプションです。

- `paper=[<紙サイズ名>/{<寸法>,<寸法>}]` : 紙サイズです。紙サイズ名は a0 から a10, b0 から b10, c2 から c8 を指定できます。B 列は JIS B 列です。また、{<横>,<縦>} と直接寸法を指定することもできます。
- `fontsize=<寸法;Q>` : 欧文フォントサイズ。デフォルトは 10pt。
- `jafontsize=<寸法;Q>` : 和文フォントサイズ。
- `jafontscale=<実数値>` : 欧文フォントと和文フォントの比 (和文/ 欧文)。 `fontsize` と `jafontsize` が両方指定されている場合は無視される。デフォルトは 1。
- `line_length=<寸法;zw,zh>` : 一行の長さ。デフォルトは字送り方向の紙幅の 0.75 倍。実際の値は一文字の長さの整数倍になるように補正されます。
- `number_of_lines=<自然数値>` : 一ページの行数。デフォルトは行送り方向の紙幅の 0.75 倍になるような値。
- `gutter=<寸法;zw,zh>` : のどの余白の大きさ。
 - `tate` 無指定時は奇数ページ左, 偶数ページ右の余白
 - `tate` 指定時は奇数ページ右, 偶数ページ左の余白
 - `twoside` が指定されていない時は、常に奇数ページ扱いで余白が設定される
- `head_space=<寸法;zw,zh>` : 天の空き量。デフォルトは中央寄せになるような値。
- `foot_space=<寸法;zw,zh>` : 地の空き量。デフォルトは中央寄せになるような値。
- `baselineskip=<寸法;Q,zw,zh>` : 行送り。デフォルトは `jafontsize` の 1.7 倍。
- `linegap=<寸法;Q,zw,zh>` : 行間。

- `headfoot_sidemargin=<寸法;zw,zh>`：柱やノンブルの左右の空き.
- `column_gap=<寸法;zw,zh>`：段間 (`twocolumn` 指定時のみ).
- `sidenote_length=<寸法;zw,zh>`：傍注の幅を指定します.

5.2 組み方

クラスオプションです.

- `open_bracket_pos=[zenkaku_tentsuki/zenkakunibu_nibu/nibu_tentsuki]`：始め括弧が行頭にきた際の配置方法を指定します. それぞれ段落開始全角折り返し行頭天付き (デフォルト), 段落開始全角二分折り返し行頭二分, 段落開始二分折り返し行頭天付きを意味します.
- `hanging_punctuation`：ぶら下げ組をします.

5.3 逆ノンブル

クラスオプションです.

- `use_reverse_pagination`：逆ノンブルの機能を利用可能にします. `jlreqreversepage` という「読み取り専用のカウンタ」が定義されます. (本物のカウンタではありません.) `\arabic` などの命令や `\value` が適用可能です. また `\thejlreqreversepage` が `\arabic{jlreqreversepage}` として定義されます.

5.4 注関係

`\jlreqsetup` で指定します.

- `reference_mark=[inline/interlinear]`：合印の配置方法を指定します. `inline` にすると該当項目の後ろの行中に配置します. `interlinear` を指定すると該当項目の上 (横組) または右 (縦組) に配置します.
- `footnote_second_indent=<寸法>`：脚注 (横書き時) または傍注 (縦書き時) の二行目以降の字下げ量を指定します. 一行目からの相対字下げ量です.
- `sidenote_type=[number/symbol]`：傍注と本文との対応の方法を指定します. `number` が規定で, 注の位置に通し番号が入り, それにより対応が示されます. `symbol` とすると, 注の位置に特定の記号が入り, また注がついている単語が強調されます.
- `sidenote_symbol=<コード>`：`sidenote_symbol=symbol` の時に, 注の位置に入る記号. デフォルト*
- `sidenote_keyword_font=<フォント設定コード>`：`sidenote_symbol=symbol` の時に, 注のついている単語のフォント指定. デフォルトは無し (強調しない)
- `endnote_second_indent=<寸法>`：後柱の二行目以降の字下げ量を指定します. 一行目からの相対字下げ量です.

- `endnote_position=[headings/paragraph/{<見出し名 1>,<見出し名 2>,...}]` : 後注の出力場所を指定します。 `headings` は各見出しの直前 (デフォルト), `paragraph` は改段落の際に出力します。 また, `endnote_position={_chapter,_section}` とすると, `\chapter` と `\section` の直前に出力します。

5.5 キャプション

図表のキャプションを `\jlrqsetup` で変更できます。 全ての設定で, 各環境ごとの設定をすることができます。 例えば `caption_font=\normalsize,table=\Large` とすると, `table` 環境内では `\Large` が適用され, そのほかの環境内では `\normalsize` が適用されます。 他の設定も同様です。

- `caption_font=<フォント設定コード>` : キャプション自身のフォントを指定します。
- `caption_label_font=<フォント設定コード>` : キャプションのラベルのフォントを指定します。
- `caption_after_label_space=<寸法>` : ラベルとキャプションの間の空きを指定します。
- `caption_label_format=<コード>` : ラベルの書式を指定します。 `caption_label_format={#1:}` のようにします。 #1 が「図 1」のような番号に置換されます。
- `caption_align=[left/right/center/bottom/top]` : キャプションの場所を指定します。

5.6 引用

`quote / quotation / verse` 環境の挙動を `\jlrqsetup` で指定できます。

- `quote_indent=<寸法>` : 字下げを指定します。 デフォルトは `2\zw` です。 一行の長さが文字サイズの整数倍になるように調整されます。
- `quote_end_indent=<寸法>` : 字上げを指定します。 デフォルトは `0\zw` です。
- `quote_beforeafter_space=<寸法>` : 前後の空きを指定します。 `quote_beforeafter_space=1\baselineski` とすると一行あきます。
- `quote_fontsize=[normalsize/small/footnotesize/scriptsize/tiny]` : フォントサイズを指定します。

5.7 箇条書き

`\jlrqsetup` で指定します。

- `itemization_beforeafter_space=<寸法>` : 箇条書きの前後の空きを指定します。 `itemization_beforeafter_space={i=<寸法>}` とするとトップレベルのみに設定を行います。 `itemization_beforeafter_space={0pt,i=10pt,ii=5pt}` とすれば, レベル 1 の箇条書きに 10pt を, レベル 2 のそれに 5pt を, それ以外には 0pt を設定します。 レベルは上記のように小文字ローマ数字で指定します。

- `itemization_itemsep=<寸法>` : 項目同士の空きを指定します.

5.8 定理環境

`\jlrqsetup` で指定します.

- `theorem_beforeafter_space=<寸法>` : 定理環境の前後の空きを指定します.

5.9 前付きなど

`\fontmatter / \mainmatter / \backmatter / \appendix` での処理を `\jlrqsetup` で指定できます.

- `frontmatter_pagebreak=[cleardoublepage/clearpage/]` : `\fontmatter` 実行時の改ページを実行する命令名を指定します. 空にすると何もしません.
- `frontmatter_counter={<カウンタ名>={value=<値>, the=<コード>, restore=[true/false]},...}` : `\fontmatter` 時でのカウンタの操作を指定します. 例えば `chapter={value=0,the={[\arabic{chapter}]}}` とすると, `chapter` カウンタの値が 0 になり, `\thechapter` が `[\arabic{chapter}]` となります. デフォルトでは `\mainmatter` 時に値と `\the<カウンタ名>` の定義を戻しますが, `restore=false` とするとこの動きが抑制されます.
- `frontmatter_heading={<見出し命令名>={<設定>},...}` : 見出し命令の動きを変更します. `\Declare***Heading` で指定できる項目の他以下を受け付けます.
 - `heading_type=[Tobira/Block/Runin/Cutin/Modify]` : 見出しの種類です. `Modify` が指定された場合は `\ModifyHeading` での変更となります.
 - `heading_level=<数値>` : 見出し命令のレベルを設定します. 指定されなかった場合は, `\fontmatter` 実行時の値が使われます. `heading_type=Modify` の時は無視されます.
 - `restore=[true/false]` : `true` が指定されると, `\mainmatter` で元の定義を復帰します. デフォルトは `true` です.
- `frontmatter_pagestyle={<ページスタイル名>[,restore=[true/false]]}` : `\fontmatter` 実行時にここで指定されたページスタイルへと切り替えます. デフォルトでは `\mainmatter` 時にもとのページスタイルに戻しますが, `restore=false` を指定すると戻しません.
- `frontmatter_pagination={<ページ番号指定>[,continuous,independent]}` : ページ番号の出力形式を, `frontmatter_pagination=roman` のように LaTeX の命令名で指定します. 更に `continuous` が指定されると通しノンブルとなります. `independent` で別ノンブルです.
- `frontmatter_precode=<コード>` : `\fontmatter` 時に最初に実行されるコードです.
- `frontmatter_postcode=<コード>` : `\fontmatter` 時に最後に実行されるコードです.

`fontmatter` を `mainmatter` や `backmatter`, `appendix` へと変えた設定も存在します. ただし,

以下のような違いがあります。

- `restore=[true/false]` は無効な設定です。
- `mainmatter_pagination` に `continuous` と `independent` は指定できません。
- `appendix_pagebreak`, `appendix_pagestyle`, `appendix_pagination` はありません。

6 見出し

新しい見出しを `\New***Heading` という命令で作ることができます (***) には見出しの種類に応じた文字列が入る)。書式はすべて

```
\New***Heading{<命令名>}{<レベル>}{<設定>}
```

となっています。また、`\Renew***Heading`, `\Provide***Heading`, `\Declare***Heading` も同時に用意されます。それぞれ

- `\Renew***Heading` : 指定した名前の命令が定義されていなければエラー。
- `\Provide***Heading` : 指定した名前の命令が定義されていない場合に限り見出し命令の定義が行われる。
- `\Declare***Heading` : 指定した名前の命令が定義されているか否かによらず新しく見出し命令を定義する。

となっています。

6.1 扉見出し

`\NewTobiraHeading` で作成します。通常のクラスファイルにおける `\section` 等と同じ書式の命令ができます。設定は以下の通り。

- `type=[han/naka]` : `han` だと半扉見出しを、`naka` だと中扉見出しを作ります。
- `pagestyle=<ページスタイル名>` : 見出し箇所のページスタイルを指定します。
- `label_format=<コード>` : ラベルを出力する命令を指定します。たとえば `label_format={第\thechapter 章}` のように指定します。
- `format=<コード>` : 実際に出力する書式を指定します。 `format={\null\vfil{\Huge\bfseries #1#2}}` のようにします。 #1 はラベルに、 #2 は見出し文字列に置き換えられます。この中では `\jlreqHeadingLabel`, `\jlreqHeadingText` という命令が利用可能です。いずれも引数を一つとる命令で、それぞれラベル、見出し文字列が空ならば空に、そうでなければ与えられた引数自身を出力します。例えば `format={[\jlreqHeadingLabel{Label=#1}]}` と指定されていれば、ラベルが空でない時には `[Label=<ラベル>]` を、そうでなければ `[]` を出力します。

- `number=[true/false]` : 採番を行うかを指定します。ただし、`number=false` の場合でも対応するカウンタは定義されます。また `\the<カウンタ名>` の変更もされないので、必要ならば再定義が必要になります。

6.2 別行見出し

`\NewBlockHeading` で作成します。 `\<命令名>*[running head]{見出し文字列}[副題]` という書式の命令を作成します。設定は以下の通り。

書式関連

- `font=<フォント設定コード>` : 見出しのフォントを指定します。
- `subtitle_font=<フォント設定コード>` : 副題のフォントを指定します。
- `label_format=<コード>` : ラベルのフォーマットを指定します。 `label_format={第 \thechapter 章}` などのようにします。
- `subtitle_format=<コード>` : 副題のフォーマットを指定します。 `subtitle_format={「#1」}` のようにします。 #1 は副題自身に置き換えられます。
- `format=<コード>` : 見出し全体のフォーマットを指定します。 #1 がラベル、 #2 が見出し文字列、 #3 が副題に置き換えられます。内部では `\jlrreqHeadingLabel`、 `\jlrreqHeadingText`、 `\jlrreqHeadingSubtitle` という命令が利用可能です。いずれも引数を一つとる命令で、それぞれラベル、見出し文字列、副題が空ならば空に、そうでなければ与えられた引数自身を出力します。例えば `format={[\jlrreqHeadingLabel{Label=#1}]}` と指定されていれば、ラベルが空でない時には `[Label=<ラベル>]` を、そうでなければ `[]` を出力します。なお、実際に #1 が置き換えられるのはラベル自身ではなく、それに空きの調整などが入ったコードです。従って、予期しない結果を得ることもあり得ます。 #2、 #3 も同様です。

インデント関連

- `align=[left/center/right]` : 見出し位置の横方向の配置場所を指定します。
- `indent=<寸法>` : 見出し全体の字下げ量を指定します。
- `end_indent=<寸法>` : 見出し全体の字上げ量を指定します。
- `after_label_space=<寸法>` : ラベル後、見出し文字列までの空きを指定します。
- `second_heading_text_indent=[<寸法>/{<寸法>,<寸法>}]` : 見出し文字列の二行目以降のインデントを指定します。見出し文字列一行目の頭を起点として指定しますが、 `second_heading_text_indent=*1\zw` のように先頭に * をつけるとラベルの頭を起点としての指定になります。また、 `second_heading_text_indent={<ラベルがある時>,<ラベルがない時>}` という指定をすると、ラベルの有無に応じて値を変更することができます。 <ラベルがある時> の指定ではやはり * を使うことができます。
- `subtitle_indent=<寸法>` : 副題のインデント量を指定します。見出し文字列の一行目を起点として指定します。ただし、 `subtitle_indent=*1\zw` のように先頭に * をつけるとラベルの

頭を起点としての指定になります。 `subtitle_break=true` の時のみ有効です。

その他

- `subtitle_break=[true/false]` : 見出し文字列と副題の間を改行するか指定します。
- `allowbreak_if_evenpage=[true/false]` : 見出しが偶数ページにあった場合、その直後の改ページを許可します。
- `pagebreak=[clearpage/cleardoublepage/clearcolumn/nariyuki/begin_with_odd_page/begin_with_even_page]` : 見出し直前の改ページを指定します。それぞれ、改ページ、`\cleardoublepage` 実行、改段、なりゆき、奇数ページ開始、偶数ページ開始、です。
- `afterindent=[true/false]` : 見出し直後の段落の字下げを行うかを指定します。
- `column_spanning=[true/false]` : 段抜きの見出しにします。 `pagebreak=nariyuki` または `pagebreak=clearcolumn` の時には無視されます。
- `number=[true/false]` : 採番を行うかを指定します。 `\NewTobiraHeading` と同様の注意が必要です。

行取り 行取りの指定は以下のいずれかの方法で行うことができます。

- 行数を指定し、その中央に配置します。 `lines=<自然数値>` により行数を指定します。 `before_lines=<自然数値>` や `after_lines=<自然数値>` により、さらに前後に追加する行数を指定します。たとえば `lines=3,after_lines=1` とすれば、四行の中に配置され、前の空きよりも後ろの空きの方が一行分大きくなります。 `before_lines` により指定された空きはページ頭には入りませんが、 `before_lines=*1` というように `*` を先頭につけると常に入るようになります。
- 行数と、前後いずれかの空きを指定します。 `lines=<自然数値>` により行数を、 `before_space=<寸法>` または `after_space=<寸法>` のいずれかの指定によりそれぞれ前または後ろの空きを指定します。
- 前後の空きを指定します。 `before_space=<寸法>` および `after_space=<寸法>` を指定します。

連続して掲げる見出しの行取り `\SetBlockHeadingSpaces` により、見出しが連続して掲げられたときの行取りを設定することができます。 `\SetBlockHeadingSpaces` は

```
\SetBlockHeadingSpaces{
  {_part{lines=3,before_lines=1},_section{lines=2},_subsection{lines=2}}
  [lines=5]{_section,23pt,_subsection,16pt}
}
```

のように使います。この意味は次の通りです。

- `\part`, `\section`, `\subsection` という順番で見出しが掲げられていて、その前後が見出し

でない場合は、`\part` は三行取り+前に一行空き、`\section` と `\subsection` は二行取りとなります。

- `\section`, `\subsection` という順番で見出しが掲げられていて、その前後が見出しでない場合は、全体で五行取りとし、`\section` と `\subsection` との間に 23pt の空き、`\subsection` の後に 16pt の空きを入れます。

個々の設定は以下のようになります。

- 各々の{}内には、`<見出し命令名>`か`<寸法>`をカンマ区切りで並べます。
- 先頭に `[]` で囲まれた設定を追加できます。これは連続して掲げられた見出し全体への設定となります。 `lines` / `before_lines` / `after_lines` / `before_space` / `after_space` が利用可能です。各々の意味は上述の行取り指定と同じです。
- 寸法はそのまま空き量を表します。
- `<見出し命令名>`の後に{}で囲まれた設定を追加することで、その見出しの空き量を設定します。設定しない場合は前後に空きが入りません。
- 見出しに対する{}で囲まれた設定内では、 `lines` / `before_lines` / `after_lines` / `before_space` / `after_space` が利用可能です。各々の意味は上述の行取り指定と同じです。
- {}で囲まれた部分を * のみにすると（例えば `\section{*}` とすると）単独で掲げた場合と同じ設定を使います。

なお、見出しが連続しているかは単純に別行見出しの命令が並んで書かれているかのみで判断します。従ってそれらの命令間に出力には関係しないような命令が挟まっていたとしても、見出しが連続して掲げられているとは判断されません。ただし、見出し命令の間に空白、改行または `\label[<オプション>]{<引数>}...{<引数>}` という形のもののみが挟まれている場合は、見出しが連続していると判断されます。

6.3 同行見出し

`\NewRuninHeading` で作成します。通常の文書クラスにおける `\section` と同様の、`<命令名>*[running head]{見出し文字列}` という書式の命令が作成されます。設定は以下の通り。

- `font=<フォント設定コード>`：見出しのフォントを指定します。
- `indent=<寸法>` 見出し文字列全体の字下げ量を指定します。
- `after_label_space=<寸法>`：ラベル後、見出し文字列までの空きを指定します。
- `label_format=<コード>`：ラベルのフォーマットを指定します。 `label_format={\theparagraph}` などのようにします。
- `number=[true/false]`：採番を行うかを指定します。`\NewTobiraHeading` と同様の注意が必要です。

6.4 窓見出し

`\NewCutinHeading` で作成します。 `\<命令名>{見出し文字列}` という書式の命令を作成します。設定は以下の通り。

- `font=<フォント設定コード>`：見出しのフォントを指定します。
- `indent=<寸法>`：見出し全体の字下げ量を指定します。
- `after_space=<寸法>`：見出しと本文との間の空きを指定します。
- `onelinemax=<寸法>`, `twolinemax=<寸法>`：見出し文字列の長さが `onelinemax` 以下ならば一行で、`twolinemax` 以下ならば二行で窓見出しを出力します。それ以上の場合は三行です。デフォルトはそれぞれ 6 文字, 20 文字の長さ。

6.5 \ModifyHeading

既に（上のどれかを使い）定義された見出し命令の設定を変更します。たとえば

```
\ModifyHeading{section}{lines=10}
```

とすると、`\section` のフォントなどの設定はそのままに、行取りのみが 10 行に変更されます。見出しの種類を変更することはできません。

6.6 \SaveHeading

見出し命令の定義を待避します。

```
\SaveHeading{section}{\restoresection} % \section の中身を \restoresection に待避。  
\RenewBlockHeading{section}{1}{font=……} % \section を新しく定義する。  
……  
\restoresection % \section の中身を元に戻す。
```

のように使います。

7 ページスタイル

```
\NewPageStyle{<ページスタイル名>}{<設定>}
```

によりページスタイルを定義することができます。<設定>は `keyval` 形式です。定義したページスタイルは `\pagestyle` で適用できます。設定は以下の通り。

- `yoko`：横書きで上下に出力します。デフォルト。
- `tate`：縦書きで小口側に出力します。
- `running_head_font=<フォント設定命令>`：柱のフォントを指定します。

- `nombre_font=<フォント設定命令>`：ノンブルのフォントを指定します。
- `running_head_position`, `nombre_position`：柱とノンブルの位置を指定します。yoko か tate のどちらかが指定されているかで指定方法が変わります。
 - yoko 指定時：top-left のように指定できます。top / bottom / center / left / right / gutter / fore-edge が使えます。gutter はのど、fore-edge は小口です。left, right の指定は奇数ページに対するものです。twoside が指定されている場合、偶数ページはその逆になります。
 - tate 指定時：<寸法>が指定できます。running_head_position は柱の天からの下げ量を、nombre_position はノンブルの地からの上げ量を指定します。
- `nombre=<書式>`：出力するノンブルを指定します。デフォルトは `\thepage`。
- `odd_running_head=<書式>`, `even_running_head=<書式>`：それぞれ奇数ページ、偶数ページの柱を指定します。_section のように_から始まる名前を指定すると、対応する見出しを出力します。（_section だと現在の \section を出力する。）
- `mark_format={ [odd=<書式>/even=<書式>/_<見出し命令名>=<書式>], ... }`：見出しを柱に出力する際のフォーマットを指定します。mark_format={_section={節 \thesection: #1},_chapter={第 \thechapter 章 \quad #1}} のように指定します。見出し命令名の代わりに odd や even も指定でき、それぞれ奇数ページ/偶数ページの柱の書式になります。pagestyle 実行時に \sectionmark 等を定義することで実現しています。
- `nombre_ii=<書式>`：二つ目のノンブルを指定します。nombre_ii_position で場所指定、nombre_ii_font でフォント設定もできます。指定方法は nombre や nombre_position と同じです。odd_running_head_ii, even_running_head_ii, running_head_ii_position, running_head_ii_font もあります。nombre_ii_position や running_head_ii_position が指定されなかった場合、yoko 指定時にはそれぞれ nombre_position および running_head_position と同じ位置に設定されます。tate 指定時は一つ目のノンブルや柱に続く場所に表示されます。

`\RenewPageStyle`, `\ProvidePageStyle`, `\DeclarePageStyle` もあります。`\ModifyPageStyle` により既存のページスタイルを改変することが可能です。

8 JFM

以下のような独自の JFM を使います。パッケージによっては、パッケージ独自の JFM や、また標準の JFM を使うように設定がし直される場合があります。本クラスファイルで使用する JFM を使う場合には、パッケージオプションなどを適切に与えて設定をする必要があります。

8.1 pLaTeX/upLaTeX の場合

JFM の名前は次の通りです。[] で囲まれている文字は設定により入ったり入らなかったりします。

```
[u] [b] [z] jlreq[g] [-v]
```

それぞれの文字は以下の場合に入ります.

- u: upLaTeX 利用時
- b: ぶら下げ組み利用時. (クラスオプションに `hanging_punctuation` が指定された時.)
- z: 行頭における開き括弧類の前の空きが, 段落開始時が全角二分, 折り返し時が二分の時. (クラスオプションに `open_bracket_pos=zenkakunibu_nibu` が指定された時.)
- g: ゴシック用フォント.
- -v: 縦書き用.

例えば, ぶら下げ組みを利用せず, クラスオプションに `open_bracket_pos=zenkakunibu_nibu` が指定されているソースを pLaTeX で処理した場合, 横書き明朝体には `zjlreq` という名前の JFM が使われます.

8.2 LuaLaTeX の場合

- 横書き用の JFM は `jlreq`
- 縦書き用の JFM は `jlreqv`

となります. ゴシックも同じ JFM を使います. 本クラスファイルは, LuaTeX-ja 標準の JFM をこれらに変更します.

9 その他

- クラスオプション `jlreq_notes` が渡されると, 日本語組版処理の記述と矛盾する設定が行われた場合に通知がされます.

10 ライセンス

このパッケージは二条項 BSD ライセンスの元で配布されています. 詳しくは [LICENSE](#) をご覧ください.

11 履歴

- 2017-02-08
 - 最初のバージョン.
- 2017-02-17
 - いくつかバグを修正.
 - クラスオプション `\jlreqsetup` にいくつかのキーを追加/変更.
 - `abstract` 環境を実装.

- パッケージを読み込んでいるだけののはやめた.
- 2017-03-14
 - いくつかバグを修正.
 - 和文ファミリを欧文ファミリに従属させるようにした.
 - `\DeclareBlockHeading` にオプションをたくさん追加.
 - `quote` 環境などを調整するオプションを追加.
- 2017-03-20
 - バグ修正.
 - `\footnote` / `\sidenote` / `\endnote` の周りに必要ならば空白を挿入するようにした.
- 2017-04-04
 - バグ修正.
 - `\DeclarePageStyle` に `tate` と `font` オプションを追加.
- 2017-04-29
 - バグ修正
 - `jafontsize` と `jfontscale` をクラスオプションに, また `\jafontsize` を追加.
 - `\tatechuyoko` を追加.
 - クラスオプション `jlreq_warnings` を `jlreq_notes` に変更.
 - いくつかのクラスオプションを `\jlreqsetup` に移動.
 - いくつかのオプションを `\jlreqsetup` に追加.
 - クラスオプションの `paper={<縦>, <横>}` を `paper={<横>, <縦>}` に変更.
- 2017-06-11
 - `plex` / `lltjtext` の読み込みを中止.
 - `\DeclareBlockHeading` に `align` を追加. `indent=center` や `end_indent=center` を廃止.
 - 一部の `\kcatcode` (upLaTeX 時) を変更.
- 2017-08-13
 - `column_spanning` を `\DeclareBlockHeading` に追加.
 - ページレイアウトにおける「本文の長さ」に傍注の長さを入れるようにした.
 - 傍注の長さのデフォルトを 0 とした.
 - 傍注の長さが 0 の時には `\sidenote` を定義しないようにした.
 - 和字間隔を挿入する命令を追加.
- 2017-08-29
 - 縦書きでも著者名が横書きで出てしまうバグを修正.
- 2017-11-23
 - バグ修正
 - `\SetBlockHeadingSpaces` を追加.
 - `\contentsname` と `\indexname` に入っていたスペースを削除.

- 2017-12-02
 - バグ修正
- 2017-12-22
 - JFM を改善.
 - 別行見出しの間の `\label` の検出方法を変更.
 - `\theequation`, `\thefigure`, `\thetable` に章番号を追加.
- 2018-02-01
 - 縦書きの傍注は奇数ページにのみ出るようにした (改善の余地ありかも).
 - LuaTeX 時に `\fnfixbottomtrue` を追加.
 - キャプション関係のオプションを `\jreqsetup` に追加.
 - `itemization_beforeafter_space` を拡張.
 - バグ修正.
- 2018-04-11
 - 縦書き二段組みの傍注を下段に出すようにした.
 - `begin_width_(oddeven)_page` を `\DeclareBlockHeading` に追加.
 - `\labelenumi` らを `jarticle` などに合わせた.
 - `column_gap` クラスオプションを使うとコンパイルできなかったバグ修正.
 - `mark_format` を `\DeclarePageStyle` に追加.
- 2018-05-19
 - 目次内のラベルの長さを今までよりも長くした.
 - 一部のマクロを `jreq-helpers.sty` に分離した.
 - バグ修正.
- 2018-06-17
 - シリーズ `b` もゴシックにするようにした.
 - バグ修正.
- 2018-08-08
 - `\DeclarePageStyle` に `nombre_ii` 等を追加.
 - バグ修正.
 - `\jreqsetup` に `footnote_second_indent` と `endnote_second_indent` を追加.
- 2018-08-15
 - バグ修正.
- 2018-09-01
 - `\mag` が 1000 でない場合も動くようにした (つもり).
 - バグ修正.
- 2018-12-10
 - 見出し命令を作る命令に `number=[true/false]` を追加.
 - `\frontmatter` 等の挙動を設定できるようにした.

- \jlreqHeadingLabel 等を扉見出しと別行見出しの format 内で使えるようにした.
- バグ修正
- 2019-01-15
 - \NewPageStyle に nombre_font などを追加, font もまだ有効だが以降非推奨とする.
 - \NewBlockHeading の format に #1 が含まれてもエラーが起こらないようにした.
 - \jlreqsetup の caption_label_format などを拡張.
 - バグ修正.
- 2019-04-01
 - 逆ノンブルを利用可能にするクラスオプション use_reverse_pagination を追加.
 - zref パッケージの利用をやめた.
 - 新元号を追加.
 - バグ修正.
- 2019-05-07
 - わずかな誤差で行数が減ってしまうのを防ぐために, \textwidth と \textheight を少しだけ増やした.
 - \DeclarePageStyle 内の running_head_ii などの実装を変更.
 - バグ修正

Noriyuki Abe <https://github.com/abenori/jlreq>