

# JLReq 文書クラス

## 概要

1

二段組文書で通常の **概要** 環境を使用すると、概要は一段中に収まる形で表示されます。二段組み文書で、概要を二段にまたがって表示するには、この例のように `Abstract over Columns` 環境を指定してください（この文書は一段組になっていますので、どちらを使っても同じです）。この `Abstract over Columns` 環境<sup>(1)</sup>は、プリアンブル中に `LaTeX` コマンドの形で渡されますので、一定の制約があります。とくに改段落は `\par` コマンドを `TeX` コードで渡す必要があります。このように→  
また、プリアンブル中に宣言される変数等は、`Abstract over Columns` 環境中では使用できないものがあります。使用できるかどうかは、プリアンブル中の宣言順序に依存します。宣言順序は `LATEX` で指定することはできません（天に祈ってください）。なお、下記の側註は `sidenote_type=symbol` が宣言される前にあるので、側註タイプが記号ではなく番号になっています<sup>(2)</sup>。

## 文書設定

—まずは足場作りから—

JLReq 文書クラスには、`Japanese Article (JLReq Class)`・`Japanese Book (JLReq Class)`・`Japanese Report (JLReq Class)` の三つのクラスがあり、どれを選ぶかによって、`article`, `book`, `report` クラス相当の出力を選択することができます。文書内の各種設定は、オプションの種類によって

(1) 縦書き文書で概要を使うことはあまりないかもしれませんが、この文書は縦書きのみで使用可能な機能を例示するために、縦書きの設定にしています。

(2) この挙動をコントロールする必要がある場合は、この環境を使わず、プリアンブル中に直接コードを書いてください。

- クラスオプションで渡すか (文書 ▽ 設定 ▽ 文書クラス ▽ クラスオプション ▽ 詳細設定 にコマ区切りで指定)

- JLRq Setup 環境を使って渡すか

のいずれかで行います。文書レイアウト全体 (基本版面) の設定は前者の方法, その他は後者の方法で行います (詳しくは付録参照)。

下記の段落\*では, jlrq 文書クラスのグローバルな設定を行っています。この段落の内容は, `LATEX` ソース中において, プリアンブルに `\JLRqsetup{...}` の形で渡されます。これは文書中のどこにおいても構いませんし, 複数回文書中に登場しても構いません。下記の設定は, 側註が連番ではなく, 記号でマークされるように指定します。これによって, 側註差込枠の挙動が変わります (引数を取るようになります)。この指定をしない場合には, 側註差込枠は引数を取りません。また, JLRq Setup 環境を使わずに, `\JLRqsetup{...}` を使って, プリアンブル (文書 ▽ 設定 ▽ `LATEX` プリアンブル) に直接書いても同じです。

## jlrq 文書に特有な特別差込枠

**後註** 後註とは, 文書の章割りの最後 (1) に出力される註釈のことです。この文書では, 後註の内容はこの節の最後に表示されます。

**側註** 側註は余白\*部分に出力する註釈です。グローバルに定義されている傍註と同じ `\marginpar` を使用していますが, 文書クラスの体裁に合わせてある, こちらの側註の方を使ってください。縦書き文書ときには脚註として機能します。

**脚註** 脚註は横書き文書では用紙下部に出力されますが, 縦書き文書ではページの一側左側に出力さ

下記の段落  
JLRq Setup

環境の設定は,  
`LATEX` 中においてのみ表示されます。出力中には表示されません。

余白 側註はこの余白部分に出力されます。

れます(2)。

**割註** 割註とは、本文中に小文字で註釈を、かっこで括った複数行で入れるものです。

**Varichu** 特別差込枠を使用した場合には、割註の長さは自動的に計算されます(自動的に計算されるとういうことです)。一方、**Varichu\***を使用すると、区切りを手動で指定すること(ここでは長さは特段指定していません)ができます。

**Varichu\***の中身は、**LaTeX**の表の書き方に準拠して書きます(つまりはとういうことです。縦方向の区切りは二重バックスラッシュです)。 **Varichu\***には、差込枠の中身がそのまま **LaTeX**に渡されるので、特殊文字は必要に応じてエスケープしてください。

### 縦中横

縦中横とは、縦書きの中によ横書きを入れることです。 **LyX** のように短い英文を入れるのに

いいかもしれません。特別差込枠 **Tatechuyoko** は **LaTeX** コマンド **\tatechuyoko\*** を使用します。

**LaTeX** コマンド **\tatechuyoko** は、本レイアウトでは実装されていません。

**字取り** これが「字 取 り」の例です。2cm の長さに文字を当てはめています。

**空き組** 空き組は、「このようにして」入れます。ここでは、文字間に2mm の空きが入っています。これは **LuaLaTeX** での動作しません。

(1) ここが本節の最後です。後註はここに出力されます。

## この行は節見出しです

前節中に入れた後註は、右の見出しの前に出力されます。

(2) 縦書き文書での「脚註」の出力位置はここです。

## 付録 文書設定関連のオプション

文書の設定に関するオプションの一覧です。jiegの付属文書からの抜粋です。

### クラスオプションで渡すオプション

ここに列挙されたオプションは、**文書▽設定▽文書クラス▽クラスオプション▽詳細設定にコンマ区切りで記入して指定します。**

#### 標準的オプション

4

- `oneside / twoside / onecolumn / twocolumn / titlepage / notitlepage / draft / final / landscape / openright / openany / leqno / fleqn`
- `disablejfam` : 和文フォントを数式用に登録しません。

#### 基本版面に関するもの

- `paper=[<紙サイズ名>,<寸法>,<寸法>]` : 紙サイズです。紙サイズ名は `a0paper` から `a10paper`, `b0paper` から `b10paper`, `c2paper` から `c8paper` を指定できます。B列はISO B列です。JIS B列を指定する場合は、`b0j` から `b10j` の対応するものを指定してください。また、`letterpaper`, `legalpaper`, `executivepaper` が指定できます。さらに、`(<横>,<縦>)`と直接寸法を指定する`(<横>,<縦>)`。
- `fontsize=<寸法>;q,H` : 欧文フォントサイズ。デフォルトは `10pt`。

- `jafontsize=<寸法;q,H>`: 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
- `fore-edge=<寸法;zw,zh>`: 小口(のどでない方)の余白の大きさ. 「日本語組版処理の要件」にある方法で余白を指定する限り使われることはありませんが, 便利なこともあるので実装されています.
- `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,H,zw,zh>`: 行送り. デフォルトは `jafontsize` の 1.7 倍.
- `linegap=<寸法;q,H,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>`: 傍註の幅を指定します.

## 組み方に関するもの

- `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。

## 逆ノンブルに関するもの

- `use_reverse_pagination` : 逆ノンブルの機能を利用可能にします。`jltreversepage` 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。

## JLREQ Setup 環境で設定するオプション

ここに列挙されたオプションは、JLReq Setup 環境に直に書き込んで指定します。複数の指定を同一行に書き込む場合には、カンマで区切ります。JLReq Setup 環境では、書き込んだ内容がそのまま `LATEX` に渡されるので、`LATEX` において特別な意味を持つ文字を使用する際にはエスケープしてください。

## 註に関するもの

- `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>: 脚註(横書き時)` または `傍註(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]: 傍註と本文との対応の方法を指定します。` `number` が規定で、註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、註の位置に特定の記号が入り、また註がついている単語が強調されます。

- `sidenote_symbol=<コード>: sidenote_type=symbol` の時に、註の位置に入る記号。デフォルト \*

- `sidenote_keyword_font=<フォント設定コード>: sidenote_type=symbol` の時に、註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>,...}]`: 後註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}>{<章>,<節>}` \chapter >\section の直前に出力します。^見出し名^を指定するためには、対象の見出しが本クラスファイルの機能を使って作られていなければいけません。

#### キャプションに関するもの

- `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=<フォーマット>`: 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。  
`{center,*left}` のようにする。通常は中央配置だがキャプションが大きいときには左に配置されます。

#### 引用に関するもの

- `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\baselineskip` だと各行一行あかあかします。
- `quote_fontsize={normalsize/small/footnotesize/scriptsizes/tiny}`: フォントサイズを指定します。

#### 箇条書きに関するもの

- `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、レベル一の箇条書きに `10pt`



を、レベル二のそれに 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。

#### 定理環境に関するもの

- `theorem_beforeafter_space=<寸法>::` 定理環境の前後の空きを指定します。
- `theorem_label_font=<フォント設定コード>::` 定理環境のラベル部分のフォントを設定します。
- `theorem_font=<フォント設定コード>::` 定理環境本体のフォントを設定します。

#### 前付け／本文部分／後付け／付録に関するもの

以下の各オプションは、`frontmatter` 部分を `mainmatter`／`backmatter`／`appendix` に変えることによって、本文部分／後付け／付録に関する設定に変えることができます。

- `frontmatter_pagebreak=[cleardoublepage/clearpage]`：`\frontmatter` 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=<ルール>, restore=[true/false]],...}`：`\frontmatter` 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` で指定され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` 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、`\frontmatter` 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 が指定されるを、`\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]]}` : `\frontmatter` 実行時に `\page` 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` 時にもこ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` が指定されるは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>` : `\frontmatter` 時に最初に実行されるコードです。

- `\frontmatter_postcode=<コード>` : `\frontmatter` 時に最後に実行されるコードです。

`\frontmatter_` や `\mainmatter_` や `\backmatter_`、`appendix_` へと変えた場合、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` は `continuous` や `independent` は指定できません。
- `appendix_pagebreak`、`appendix_pagestyle`、`appendix_pagination` はありません。