

WORD LuaL^AT_EX Class 1.1?

Satoru KAWAHARA

2016/7/8

目次

1	はじめに	1
2	LuaT _E X-ja の読み込み	1
3	用紙サイズ	1
4	オプション	1
5	和文フォントの変更	4
6	フォントサイズ	6
7	レイアウト	9
7.1	ページレイアウト	10
8	ページスタイル	13
9	文書のマークアップ	15
9.1	表題	15
9.2	章・節	15
9.3	リスト環境	24
9.4	パラメータの設定	28
9.5	フロート	29
9.6	キャプション	30
10	フォントコマンド	31
11	相互参照	31
11.1	目次の類	31
11.2	参考文献	35
11.3	索引	36

11.4	脚注	38
12	段落の頭へのグルー挿入禁止	39
13	初期設定	41

1 はじめに

これは `ltjscssclasses.dtx` を WORD 用に改変したものです。次のドキュメントクラス（スタイルファイル）を生成します。元となった `ltjscssclasses.dtx` と同様に、このクラスファイルのライセンスは修正 BSD ライセンスとします。

オプション	意味
<code><word-lua></code>	word-lua クラスを生成

以下では実際のコードに即して説明します。

2 LuaTeX-ja の読み込み

まず、`luatexja` を読み込みます。

```
1 \RequirePackage{luatexja}
```

3 用紙サイズ

用紙サイズを指定します。WORD は JIS B5 です。

```
2 \AtBeginDvi{\special{papersize=182mm,257mm}}
3 % \end{macrocode}
4 %
5 % \begin{macrocode}
6 \setlength\paperheight {257mm}
7 \setlength\paperwidth {182mm}
```

4 オプション

これらのクラスは `\documentclass{ltjsarticle}` あるいは `\documentclass[オプション]{ltjsarticle}` のように呼び出します。

まず、オプションに関連するいくつかのコマンドやスイッチ（論理変数）を定義します。

`\if@restonecol` 段組のときに真になる論理変数です。

```
8 \newif\if@restonecol
```

`\if@titlepage` これを真にすると表題、概要を独立したページに出力します。

```
9 \newif\if@titlepage
```

`\if@mainmatter` 真なら本文、偽なら前付け・後付けです。偽なら `\chapter` で章番号が出ません。

```
10 \newif\if@mainmatter \@mainmattertrue
```

`\if@enablejfam` 和文フォントを数式フォントとして登録するかどうかを示すスイッチですが、実際には用いられません。太古のスタイルファイルとの互換性のために残されています。

```
11 \newif\if@enablejfam \@enablejfamtrue
```

以下で各オプションを宣言します。

■ヘッダー表示位置のスイッチ `swapheader` で、奇数ページの subtitle を左に、偶数ページの subtitle を右に出力する事ができます。

```
12 \newif\if@swapheader
```

```
13 \@swapheaderfalse
```

```
14 \DeclareOption{swapheader}{\@swapheadertrue}
```

■両面、片面オプション `twoside` で奇数ページ・偶数ページのレイアウトが変わります。

[2003-04-29] `var twoside` でどちらのページも傍注が右側になります。

```
15 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
```

```
16 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
```

```
17 \DeclareOption{var twoside}{\@twosidetrue \@mparswitchfalse}
```

■二段組 `twocolumn` で二段組になります。

```
18 \DeclareOption{onecolumn}{\@twocolumnfalse}
```

```
19 \DeclareOption{twocolumn}{\@twocolumntrue}
```

■`eqnarray` 環境と数式の位置 森本さんのご教示にしたがって前に移動しました。

`eqnarray` L^AT_EX の `eqnarray` 環境では `&` でできるアキが大きすぎるようですので、少し小さくします。また、中央の要素も `\displaystyle` にします。

```
20 \def\eqnarray{%
```

```
21   \stepcounter{equation}%
```

```
22   \def\@currentlabel{\p@equation\theequation}%
```

```
23   \global\@eqnswtrue
```

```
24   \m@th
```

```
25   \global\@eqcnt\z@
```

```
26   \tabskip\@centering
```

```
27   \let\\\@eqnocr
```

```
28   $$\everycr{}\halign to\displaywidth\bgroup
```

```
29     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
```

```
30     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
```

```
31     &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering
```

```
32     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
```

```
33     \tabskip\z@skip
```

```
34     \cr}
```

`leqno` で数式番号が左側になります。`fleqn` で数式が本文左端から一定距離のところに出力されます。森本さんにしたがって訂正しました。

```
35 \DeclareOption{leqno}{\input{leqno.clo}}
```

```

36 \DeclareOption{fleqn}{\input{fleqn.clo}}
37 % fleqn 用の eqnarray 環境の再定義
38 \def\eqnarray{%
39   \stepcounter{equation}%
40   \def\@currentlabel{\p@equation\theequation}%
41   \global\@eqnswtrue\m@th
42   \global\@eqcnt\z@
43   \tabskip\mathindent
44   \let\=\@eqnocr
45   \setlength\abovedisplayskip{\topsep}%
46   \ifvmode
47     \addtolength\abovedisplayskip{\partopsep}%
48   \fi
49   \addtolength\abovedisplayskip{\parskip}%
50   \setlength\belowdisplayskip{\abovedisplayskip}%
51   \setlength\belowdisplayshortskip{\abovedisplayskip}%
52   \setlength\abovedisplayshortskip{\abovedisplayskip}%
53   $$\everycr{}\halign to\linewidth% $$
54   \bgroup
55     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnsele
56     &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
57     &\global\@eqcnt\tw@
58     $\displaystyle{##}$\hfil \tabskip\@centering
59     &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
60   \tabskip\z@skip\cr
61   }}

```

■数式フォントとして和文フォントを登録しないオプション pTeX では数式中では 16 通りのフォントしか使えませんでした、 LuaTeX では Omega 拡張が取り込まれていて 256 通りのフォントが使えます。ただし、LaTeX 2_ε カーネルでは未だに数式ファミリの数は 16 個に制限されているので、実際に使用可能な数式ファミリの数を増やすためには lualatex-math パッケージを読み込む必要があることに注意が必要です。

```

62 \DeclareOption{disablejfam}{%
63   \ClassWarningNoLine{\@currname}{The class option 'disablejfam' is obsolete}}

```

■ドラフト draft で overfull box の起きた行末に 5pt の罫線を引きます。

```

64 \newif\ifdraft
65 \DeclareOption{draft}{\drafttrue \setlength\overfullrule{5pt}}
66 \DeclareOption{final}{\draftfalse \setlength\overfullrule{0pt}}

```

■和文フォントメトリックの選択 ここでは OTF パッケージのメトリックを元とした、jfm-ujis.lua メトリックを標準で使います。古い min10, goth10 互換のメトリックを使いたいときは mingoth というオプションを指定します。pTeX でよく利用される jis フォントメトリックと互換のメトリックを使いたい場合は, ptexjis というオプションを指定します。winjis メトリックは用済みのため、winjis オプションは無視されます。

```

67 \newif\ifmingoth

```

```

68 \mingothfalse
69 \newif\ifjisfont
70 \jisfontfalse
71 \newif\ifptexjis
72 \ptexjisfalse
73 \DeclareOption{winjis}{%
74   \ClassWarningNoLine{\currname}{The class option 'winjis' is obsolete}}
75 \DeclareOption{uplatex}{%
76   \ClassWarningNoLine{\currname}{The class option 'uplatex' is obsolete}}
77 \DeclareOption{mingoth}{\mingothtrue}
78 \DeclareOption{ptexjis}{\ptexjistruer}
79 \DeclareOption{jis}{\jisfonttrue}

```

■オプションの実行 デフォルトのオプションを実行します。

```

80 \ExecuteOptions{twoside,onecolumn,final}
81 \ProcessOptions

```

■基準となる行送り

\n@baseline 基準となる行送りをポイント単位で表したものです。

```

82 \def\n@baseline{16}

```

■PDF の用紙サイズの設定

\luafunction101 出力の PDF の用紙サイズをここで設定しておきます。tombow が真のときは 2 インチ足し
 \luafunction102 ておきます。

```

83 \setlength{\@tempdima}{\paperwidth}
84 \setlength{\@tempdimb}{\paperheight}
85 \iftombow
86   \advance \@tempdima 2in
87   \advance \@tempdimb 2in
88 \fi
89 \setlength{\luafunction101}{\@tempdima}
90 \setlength{\luafunction102}{\@tempdimb}

```

5 和文フォントの変更

JIS の 1 ポイントは 0.3514mm (約 1/72.28 インチ), PostScript の 1 ポイントは 1/72 インチですが, TeX では 1/72.27 インチを 1pt (ポイント), 1/72 インチを 1bp (ビッグポイント) と表します。QuarkXPress などの DTP ソフトは標準で 1/72 インチを 1 ポイントとしますが, 以下ではすべて 1/72.27 インチを 1pt としています。1 インチは定義により 25.4mm です。

pTeX (アスキーが日本語化した TeX) では, 例えば従来のフォントメトリック min10 や JIS フォントメトリックでは「公称 10 ポイントの和文フォントは, 実際には 9.62216pt で出力される (メトリック側で 0.962216 倍される)」という仕様になっています。一方,

LuaTeX-jā の提供するメトリックでは、そのようなことはありません。公称 10 ポイントの和文フォントは、10 ポイントで出力されます。

この `ltjclasses` でも、派生元の `jclasses` と同じように、この公称 10 ポイントのフォントをここでは 13 級に縮小して使うことにします。そのためには、 $13\text{ Q}/10\text{ pt} \simeq 0.924872$ 倍すればいいことになります。

`\ltj@stdmcfont`, `\ltj@stdgtfont` による、デフォルトで使われ明朝・ゴシックのフォントの設定に対応しました。この 2 つの命令の値はユーザが日々の利用でその都度指定するものではなく、何らかの理由で非埋め込みフォントが正しく利用できない場合にのみ `luatexja.cfg` によってセットされるものです。

```

91 \expandafter\let\csname JY3/mc/m/n/10\endcsname\relax
92 \ifmingoth
93   \DeclareFontShape{JY3}{mc}{m}{n}{<-> s * [0.924872] \ltj@stdmcfont:jfm=min}{}
94   \DeclareFontShape{JY3}{gt}{m}{n}{<-> s * [0.924872] \ltj@stdgtfont:jfm=min}{}
95 \else
96   \ifptexjis
97     \DeclareFontShape{JY3}{mc}{m}{n}{<-> s * [0.924872] \ltj@stdmcfont:jfm=jis}{}
98     \DeclareFontShape{JY3}{gt}{m}{n}{<-> s * [0.924872] \ltj@stdgtfont:jfm=jis}{}
99   \else
100     \DeclareFontShape{JY3}{mc}{m}{n}{<-> s * [0.924872] \ltj@stdmcfont:jfm=ujis}{}
101     \DeclareFontShape{JY3}{gt}{m}{n}{<-> s * [0.924872] \ltj@stdgtfont:jfm=ujis}{}
102   \fi
103 \fi

```

これにより、公称 10 ポイントの和文フォントを 0.924872 倍したことにより、約 9.25 ポイント、DTP で使う単位 (1/72 インチ) では 9.21 ポイントということになり、公称 10 ポイントといっても実は 9 ポイント強になります。

和文でイタリック体、斜体、サンセリフ体、タイプライタ体の代わりにゴシック体を使うことにします。

```

104 \DeclareFontShape{JY3}{mc}{bx}{n}{<->ssub*gt/m/n}{}
105 \DeclareFontShape{JY3}{gt}{bx}{n}{<->ssub*gt/m/n}{}
106 \DeclareFontShape{JY3}{mc}{m}{it}{<->ssub*mc/m/n}{}
107 \DeclareFontShape{JY3}{mc}{m}{sl}{<->ssub*mc/m/n}{}
108 \DeclareFontShape{JY3}{mc}{m}{sc}{<->ssub*mc/m/n}{}
109 \DeclareFontShape{JY3}{gt}{m}{it}{<->ssub*gt/m/n}{}
110 \DeclareFontShape{JY3}{gt}{m}{sl}{<->ssub*gt/m/n}{}
111 \DeclareFontShape{JY3}{mc}{bx}{it}{<->ssub*gt/m/n}{}
112 \DeclareFontShape{JY3}{mc}{bx}{sl}{<->ssub*gt/m/n}{}
113 % \DeclareFontShape{JT3}{mc}{bx}{n}{<->ssub*gt/m/n}{}
114 % \DeclareFontShape{JT3}{gt}{bx}{n}{<->ssub*gt/m/n}{}
115 %% \DeclareFontShape{JT3}{mc}{m}{it}{<->ssub*mc/m/n}{}
116 %% \DeclareFontShape{JT3}{mc}{m}{sl}{<->ssub*mc/m/n}{}
117 %% \DeclareFontShape{JT3}{mc}{m}{sc}{<->ssub*mc/m/n}{}
118 %% \DeclareFontShape{JT3}{gt}{m}{it}{<->ssub*gt/m/n}{}
119 %% \DeclareFontShape{JT3}{gt}{m}{sl}{<->ssub*gt/m/n}{}
120 %% \DeclareFontShape{JT3}{mc}{bx}{it}{<->ssub*gt/m/n}{}
121 %% \DeclareFontShape{JT3}{mc}{bx}{sl}{<->ssub*gt/m/n}{}

```

```

122 \renewcommand\jttdefault{\gtdefault}
123 \DeclareRobustCommand\rmfamily
124     {\not@math@alphabet\rmfamily\mathrm
125     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
126 \DeclareRobustCommand\sffamily
127     {\not@math@alphabet\sffamily\mathsf
128     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
129 \DeclareRobustCommand\ttfamily
130     {\not@math@alphabet\ttfamily\mathtt
131     \romanfamily\ttdefault\kanjifamily\jttdefault\selectfont}

```

Lua \TeX -ja では和文組版に伴うグルーはノードベースで挿入するようになり、また欧文・和文間のグルーとイタリック補正は干渉しないようになりました。まだ「和文の斜体」については Lua \LaTeX カーネル側でまともな対応がされていませんが、`jsclasses.dtx` で行われていた `\textmc`, `\textgt` の再定義は不要のように思われます。

```

132 \AtBeginDocument{%
133   \reDeclareMathAlphabet{\mathrm}{\mathrm}{\mathmc}
134   \reDeclareMathAlphabet{\mathbf}{\mathbf}{\mathgt}}%

```

「 \TeX !」「 $\overline{515}$ 」の記号と数字の間に四分アキが入らないようにします。

```

135 \ltjsetparameter{jaxspmode={`!,2}}
136 \ltjsetparameter{jaxspmode={`\overline{,1}}}

```

「C や C++ では……」と書くと、C++ の直後に四分アキが入らないのでバランスが悪くなります。四分アキが入るようにしました。% の両側も同じです。

```

137 \ltjsetparameter{alxspmode={`+,3}}
138 \ltjsetparameter{alxspmode={`\%,3}}

```

\@ 欧文といえば、 \LaTeX の `\def\@{\spacefactor\@m}` という定義 ($\@m$ は 1000) では `I watch TV\@.` と書くと V とピリオドのペアカーニングが効かなくなります。そこで、次のような定義に直し、`I watch TV.\@` と書くことにします。

```

139 \def\@{\spacefactor3000\space}

```

6 フォントサイズ

フォントサイズを変える命令 (`\normalsize`, `\small` など) の実際の挙動の設定は、三つの引数をとる命令 `\@setfontsize` を使って、たとえば

```
\@setfontsize{\normalsize}{10}{16}
```

のように行います。これは

```
\normalsize
```

は 10 ポイントのフォントを使い、行送りは 16 ポイントである

という意味です。ただし、処理を速くするため、以下では 10 と同義の \LaTeX の内部命令 `\@xpt` を使っています。この `\@xpt` の類は次のものがあり、 \LaTeX 本体で定義されています。

<code>\@vpt</code>	5	<code>\@vipt</code>	6	<code>\@viipt</code>	7
<code>\@viipt</code>	8	<code>\@ixpt</code>	9	<code>\@xpt</code>	10
<code>\@xipt</code>	10.95	<code>\@xiipt</code>	12	<code>\@xivpt</code>	14.4

これらのグルーをもっても行分割ができない場合は、`\emergencystretch` に訴えます。

```
140 \emergencystretch 3\zw
```

`\ifnarrowbaselines` 欧文用に行間を狭くする論理変数と、それを真・偽にするためのコマンドです。

```
\narrowbaselines 141 \newif\ifnarrowbaselines
\narrowbaselines 142 \def\narrowbaselines{%
\narrowbaselines 143 \narrowbaselinestrue
144 \skip0=\abovedisplayskip
145 \skip2=\abovedisplayshortskip
146 \skip4=\belowdisplayskip
147 \skip6=\belowdisplayshortskip
148 \@currsize\selectfont
149 \abovedisplayskip=\skip0
150 \abovedisplayshortskip=\skip2
151 \belowdisplayskip=\skip4
152 \belowdisplayshortskip=\skip6\relax}
153 \def\widebaselines{\narrowbaselinesfalse\@currsize\selectfont}
```

`\normalsize` 標準のフォントサイズと行送りを選ぶコマンドです。

```
154 \renewcommand{\normalsize}{%
155 \@setfontsize\normalsize{8.5}{17}%
156 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
157 \abovedisplayshortskip \z@ \@plus3\p@
158 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
159 \belowdisplayskip \abovedisplayskip
160 \let\@listi\@listI}
```

ここで実際に標準フォントサイズで初期化します。

```
161 \normalsize
```

`\Cht` 基準となる長さの設定をします。11tjfont.sty で宣言されているパラメータに実際の値を

`\Cdp` 設定します。

```
\Cwd 162 \setbox0\hbox{\char\eut"A1A1}%
163 \setlength\Cht{\ht0}
\Cvs 164 \setlength\Cdp{\dp0}
\Chs 165 \setlength\Cwd{\wd0}
166 \setlength\Cvs{\baselineskip}
167 \setlength\Chs{\wd0}
```

`\small` `\small` も `\normalsize` と同様に設定します。

```
168 \newcommand{\small}{%
169 \@setfontsize\small{7.5}{14}%
170 \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
171 \abovedisplayshortskip \z@ \@plus2\p@
```



```

172 \belowdisplayskip 4\p@ \@plus2\p@ \@minus2\p@
173 \def\@listi{\leftmargin\leftmargini
174         \topsep 4\p@ \@plus2\p@ \@minus2\p@
175         \parsep 2\p@ \@plus\p@ \@minus\p@
176         \itemsep \parsep}%
177 \belowdisplayskip \abovedisplayskip}

```

`\footnotesize` `\footnotesize` も同様です。

```

178 \newcommand{\footnotesize}{%
179   \@setfontsize\footnotesize\@viipt{13}%
180   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
181   \abovedisplayshortskip \z@ \@plus\p@
182   \belowdisplayskip 3\p@ \@plus\p@ \@minus2\p@
183   \def\@listi{\leftmargin\leftmargini
184           \topsep 3\p@ \@plus\p@ \@minus\p@
185           \parsep 2\p@ \@plus\p@ \@minus\p@
186           \itemsep \parsep}%
187   \belowdisplayskip \abovedisplayskip}

```

`\scriptsize` それ以外のサイズは、本文に使うことがないので、単にフォントサイズと行送りだけ変更し

`\tiny` ます。特に注意すべきは `\large` で、これは二段組のときに節見出しのフォントとして使い、
`\large` 行送りを `\normalsize` と同じにすることによって、節見出しが複数行にわたっても段間で
`\Large` 行が揃うようにします。

```

\scriptsize 188 \newcommand{\scriptsize}{\@setfontsize\scriptsize\@vpt\@viipt}
\large      189 \newcommand{\tiny}{\@setfontsize\tiny{4.25}\@vpt}
\huge      190 \newcommand{\large}{\@setfontsize\large{9.5}{19}}
\Large     191 \newcommand{\Large}{\@setfontsize\Large{10.5}{21}}
\HUGE     192 \newcommand{\LARGE}{\@setfontsize\LARGE{12.75}{25}}
           193 \newcommand{\huge}{\@setfontsize\huge{17}{28}}
           194 \newcommand{\Huge}{\@setfontsize\Huge\@xxpt{33}}
           195 \newcommand{\HUGE}{\@setfontsize\HUGE{30}{40}}

```

別行立て数式の中では `\narrowbaselines` にします。和文の行送りのままでは、行列や
場合分けの行送り、連分数の高さなどが不釣合いに大きくなるためです。

本文中の数式の中では `\narrowbaselines` にしていません。本文中ではなるべく行送り
が変わるような大きいものを使わず、行列は `amsmath` の `smallmatrix` 環境を使うのがいい
でしょう。

```

196 \everydisplay=\expandafter{\the\everydisplay \narrowbaselines}

```

しかし、このおかげで別行数式の上下のスペースが少し違っていました。とりあえず
`amsmath` の `equation` 関係は `okumacro` のほうで逃げていますが、もっとうまい逃げ道が
あればお教えてください。

見出し用のフォントは `\bfseries` 固定ではなく、`\headfont` という命令で定めること
にします。これは太ゴシックが使えるときは `\sffamily \bfseries` でいいと思いますが、
通常の中ゴシックでは単に `\sffamily` だけのほうがよさそうです。『`LaTeX 2ε` 美文書作
成入門』（1997 年）では `\sffamily \fontseries{sbc}` として新ゴ M と合わせましたが、

`\fontseries{sbc}` はちょっと幅が狭いように感じました。

```
197 % \newcommand{\headfont}{\bfseries}
198 \newcommand{\headfont}{\gtfamily\sffamily}
199 % \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}
```

7 レイアウト

■二段組

`\columnsep` `\columnsep` は二段組のときの左右の段間の幅です。元は 10pt ですが、`2\zw` にしました。このスペースの中央に `\columnseprule` の幅の罫線が引かれます。

```
200 \setlength\columnsep{2\zw}
201 \setlength\columnseprule{0\p@}
```

■段落

`\lineskip` 上下の行の文字が `\lineskiplimit` より接近したら、`\lineskip` より近づかないようにします。元は 0pt ですが 1pt に変更しました。`normal...` の付いた方は保存用です。

```
\lineskiplimit 202 \setlength\lineskip{1\p@}
\normallineskip 203 \setlength\normallineskip{1\p@}
\normallineskiplimit 204 \setlength\lineskiplimit{1\p@}
205 \setlength\normallineskiplimit{1\p@}
```

`\baselinestretch` 実際の行送りが `\baselineskip` の何倍かを表すマクロです。たとえば

```
\renewcommand{\baselinestretch}{2}
```

とすると、行送りが通常の 2 倍になります。ただし、これを設定すると、たとえば `\baselineskip` が伸縮するように設定しても、行送りの伸縮ができなくなります。行送りの伸縮はしないのが一般的です。

```
206 \renewcommand{\baselinestretch}{}
```

`\parskip` `\parskip` は段落間の追加スペースです。元は 0pt plus 1pt になっていましたが、ここでは `\parindent` ゼロにしました。`\parindent` は段落の先頭の字下げ幅です。

```
207 \setlength\parskip{\z@}
208 \setlength\parindent{\Cwd}
```

`\@lowpenalty` `\nopagebreak`, `\nolinebreak` は引数に応じて次のペナルティ値のうちどれかを選ぶようになっています。ここはオリジナル通りです。

```
\@highpenalty 209 \@lowpenalty 51
210 \@medpenalty 151
211 \@highpenalty 301
```

`\interlinepenalty` 段落中の改ページのペナルティです。デフォルトは 0 です。

```
212 % \interlinepenalty 0
```

`\brokenpenalty` ページの最後の行がハイフンで終わる際のペナルティです。デフォルトは 100 です。

```
213 % \brokenpenalty 100
```

7.1 ページレイアウト

■縦方向のスペース

`\headheight` `\topskip` は本文領域上端と本文 1 行目のベースラインとの距離です。あまりぎりぎりの値にすると、本文中に \int のような高い文字が入ったときに 1 行目のベースラインが他のページより下がってしまいます。ここでは本文の公称フォントサイズ (10pt) にします。

[2003-06-26] `\headheight` はヘッダの高さで、元は 12pt ですが、新ドキュメントクラスでは `\topskip` と等しくしていました。ところが、`fancyhdr` パッケージで `\headheight` が小さいとおかしいことになるようですので、2 倍に増やしました。代わりに、版面の上下揃えの計算では `\headheight` ではなく `\topskip` を使うことにしました。

```
214 \setlength\topskip{\Cht}
215 \setlength\headheight{20\p@}
```

`\headsep` `\headsep` はヘッダ下端と本文領域上端との距離です。

```
216 \setlength\headsep{5mm}
```

`\footskip` `\footskip` は、本文領域の下端とフッタの下端との距離です。フッタのボックスの高さを示す、`\footheight` は削除されました。

```
217 \setlength\footskip{\Cvs}
```

`\maxdepth` `\maxdepth` は本文最下行の最大の深さです。

```
218 \setlength\maxdepth{.5\topskip}
```

■本文の幅と高さ

`\fullwidth` 本文の幅が全角 40 文字を超えると読みにくくなります。そこで、書籍の場合に限って、紙の幅が広いときは外側のマージンを余分にとって全角 40 文字に押え、ヘッダやフッタは本文領域より広く取ることにします。このときヘッダやフッタの幅を表す `\fullwidth` という長さを定義します。

```
219 \newdimen\fullwidth
```

この `\fullwidth` は `article` では紙幅 `\paperwidth` の 0.76 倍を超えない全角幅の整数倍 (二段組では全角幅の偶数倍) にします。0.76 倍という数値は A4 縦置きの場合に紙幅から約 2 インチを引いた値になるように選びました。`word-lua` では紙幅から 36 ミリを引いた値にしました。

`\textwidth` 書籍以外では本文領域の幅 `\textwidth` は `\fullwidth` と等しくします。WORD では横 48 文字としています。

```
220 \setlength\fullwidth{48\Cwd}
221 \setlength\textwidth{\fullwidth}
```

`\textheight` 紙の高さ `\paperheight` は、1 インチと `\topmargin` と `\headheight` と `\headsep` と `\textheight` と `\footskip` とページ下部の余白を加えたものです。

本文部分の高さ `\textheight` は、紙の高さ `\paperheight` の 0.83 倍から、ヘッダの高さ、ヘッダと本文の距離、本文とフッタ下端の距離、`\topskip` を引き、それを `\baselineskip` の倍数に切り捨て、最後に `\topskip` を加えます。念のため 0.1 ポイント余分に加えておきます。0.83 倍という数値は、A4 縦置きの場合に紙の高さから上下マージン各約 1 インチを引いた値になるように選びました。

[2003-06-26] `\headheight` を `\topskip` に直しました。以前はこの二つは値が同じであつたので、変化はないはずです。

```
222 \setlength{\textheight}{33\cvs}
223 \addtolength{\textheight}{\topskip}
```

`\marginparsep` `\marginparsep` は欄外の書き込みと本文との間隔です。`\marginparpush` は欄外の書き込みどうしの最小の間隔です。

```
224 \setlength\marginparsep{\z@}
225 \setlength\marginparpush{\z@}
```

`\oddsidemargin` それぞれ奇数ページ、偶数ページの左マージンから 1 インチ引いた値です。片面印刷では `\evensidemargin` が使われます。TeX は上・左マージンに `1truein` を挿入しますが、トンボ関係のオプションが指定されると `lltjcore.sty` はトンボの内側に `1in` のスペース (`1truein` ではなく) を挿入するので、場合分けしています。

```
226 \setlength{\@tempdima}{\paperwidth}
227 \addtolength{\@tempdima}{-\textwidth}
228 \setlength{\oddsidemargin}{.5\@tempdima}
229 \addtolength{\oddsidemargin}{-1in}
230 \setlength{\evensidemargin}{\oddsidemargin}
```

`\marginparwidth` `\marginparwidth` は欄外の書き込みの横幅です。外側マージンの幅 (`\evensidemargin` + 1 インチ) から 1 センチを引き、さらに `\marginparsep` (欄外の書き込みと本文のアキ) を引いた値にしました。最後に `1\zw` の整数倍に切り捨てます。

```
231 \setlength\marginparwidth{\z@}
```

`\topmargin` 上マージン (紙の上端とヘッダ上端の距離) から 1 インチ引いた値です。

[2003-06-26] `\headheight` を `\topskip` に直しました。以前はこの二つは値が同じであつたので、変化はないはずです。

[2011-10-03 LTJ] ここの `\oddsidemargin` のときと同様に `-\inv@margin` ではなく `-1in` にします。

```
232 \setlength\topmargin{-1.0in}
233 \addtolength\topmargin{12mm}
```

■脚注

`\footnotesep` 各脚注の頭に入る支柱 (strut) の高さです。脚注間に余分のアキが入らないように、`\footnotesize` の支柱の高さ (行送りの 0.7 倍) に等しくします。

```
234 \setlength{\footnotesep}{3mm}
```

`\footins` `\skip\footins` は本文の最終行と最初の脚注との間の距離です。標準の 10 ポイントクラスでは 9 plus 4 minus 2 ポイントになっていますが、和文の行送りを考えてもうちょっと大きくします。

```
235 \setlength{\skip\footins}{5mm \@plus 10mm \@minus 3mm}
```

■フロート関連 フロート（図、表）関連のパラメータは \LaTeX 2_ϵ 本体で定義されていますが、ここで設定変更します。本文ページ（本文とフロートが共存するページ）ちなみに、カウンタは内部では `\c@` を名前に冠したマクロになっています。とフロートだけのページで設定が異なります。

`\c@topnumber` `topnumber` カウンタは本文ページ上部のフロートの最大数です。
[2003-08-23] ちょっと増やしました。

```
236 \setcounter{topnumber}{2}
```

`\topfraction` 本文ページ上部のフロートが占有できる最大の割合です。フロートが入りやすいように、元の値 0.7 を 0.8 [2003-08-23: 0.85] に変えてあります。

```
237 \renewcommand{\topfraction}{.7}
```

`\c@bottomnumber` `bottomnumber` カウンタは本文ページ下部のフロートの最大数です。
[2003-08-23] ちょっと増やしました。

```
238 \setcounter{bottomnumber}{1}
```

`\bottomfraction` 本文ページ下部のフロートが占有できる最大の割合です。元は 0.3 でした。

```
239 \renewcommand{\bottomfraction}{.3}
```

`\c@totalnumber` `totalnumber` カウンタは本文ページに入りうるフロートの最大数です。
[2003-08-23] ちょっと増やしました。

```
240 \setcounter{totalnumber}{3}
```

`\textfraction` 本文ページに最低限入らなければならない本文の割合です。フロートが入りやすいように元の 0.2 を 0.1 に変えました。

```
241 \renewcommand{\textfraction}{.2}
```

`\floatpagefraction` フロートだけのページでのフロートの最小割合です。これも 0.5 を 0.8 に変えてあります。

```
242 \renewcommand{\floatpagefraction}{.8}
```

`\c@dbltopnumber` 二段組のとき本文ページ上部に出力できる段抜きフロートの最大数です。
[2003-08-23] ちょっと増やしました。

```
243 \setcounter{dbltopnumber}{9}
```

`\dbltopfraction` 二段組のとき本文ページ上部に出力できる段抜きフロートが占めうる最大の割合です。0.7 を 0.8 に変えてあります。

```
244 \renewcommand{\dbltopfraction}{.8}
```

`\dblfloatpagefraction` 二段組のときフロートだけのページに入るべき段抜きフロートの最小割合です。0.5 を 0.8 に変えてあります。

```
245 \renewcommand{\dblfloatpagefraction}{.8}
```

`\floatsep` `\floatsep` はページ上部・下部のフロート間の距離です。`\textfloatsep` はページ上部・下部のフロートと本文との距離です。`\intextsep` は本文の途中に出力されるフロートと本文との距離です。

```
246 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
247 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
248 \setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

`\dblfloatsep` 二段組のときの段抜きのフロートについての値です。

```
\dbltextfloatsep 249 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
250 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

`\@fptop` フロートだけのページに入るグルーです。`\@fptop` はページ上部, `\@fpbot` はページ下部, `\@fpsep` `\@fpsep` はフロート間に入ります。

```
\@fpbot 251 \setlength\@fptop{0\p@ \@plus 1fil}
252 \setlength\@fpsep{8\p@ \@plus 2fil}
253 \setlength\@fpbot{0\p@ \@plus 1fil}
```

`\@dblfpptop` 段抜きフロートについての値です。

```
\@dblfpsep 254 \setlength\@dblfpptop{0\p@ \@plus 1fil}
255 \setlength\@dblfpsep{8\p@ \@plus 2fil}
\@dblfpbot 256 \setlength\@dblfpbot{0\p@ \@plus 1fil}
```

8 ページスタイル

`word-lua.cls` では、つぎの 2 種類のページスタイルを使用できます。このうち、`empty` は $\text{\LaTeX 2}_{\epsilon}$ 本体にて定義されています。

`empty` ヘッダにもフッタにも出力しない

`plain` ヘッダに`@subtitle`を出力する。記事標準

ページスタイルは `\ps@...` の形のマクロで定義されています。

`\@evenhead` `\@oddhead`, `\@oddfoot`, `\@evenhead`, `\@evenfoot` は偶数・奇数ページの柱（ヘッダ, フッタ）を出力する命令です。これらは `\fullwidth` 幅の `\hbox` の中で呼び出されます。`\@evenfoot` `\ps@...` の中で定義しておきます。

`\@oddfoot` 柱の内容は、`\chapter` が呼び出す `\chaptermark{何々}`, `\section` が呼び出す `\sectionmark{何々}` で設定します。柱を扱う命令には次のものがあります。

`\markboth{左}{右}` 両方の柱を設定します。

`\markright{右}` 右の柱を設定します。

`\leftmark` 左の柱を出力します。

`\rightmark` 右の柱を出力します。

柱を設定する命令は、右の柱が左の柱の下位にある場合は十分ともに動作します。たとえば左マークを `\chapter`, 右マークを `\section` で変更する場合はこれにあたります。しかし、同一ページに複数の `\markboth` があると、おかしい結果になることがあります。

`\tableofcontents` のような命令で使われる `\mkboth` は、`\ps@...` コマンド中で `\markboth` か `\gobbletwo` (何もしない) に `\let` されます。

`\ps@empty` `empty` ページスタイルの定義です。L^AT_EX 本体で定義されているものをコメントアウトした形で載せておきます。

```
257 % \def\ps@empty{%
258 %   \let\mkboth\gobbletwo
259 %   \let\@oddhead\empty
260 %   \let\@oddfoot\empty
261 %   \let\@evenhead\empty
262 %   \let\@evenfoot\empty}
```

`\ps@plain` `plain` ページスタイルの定義です。

```
263 \def\ps@plain{
264   \let\ps@jpl\in\ps@plain
265   \let\@oddfoot\empty
266   \let\@evenfoot\empty
267   \let\mkboth\markboth
268   \if@swapheader
269     \let\@evenhead\plainheaderodd
270     \let\@oddhead\plainheadereven
271   \else
272     \let\@evenhead\plainheadereven
273     \let\@oddhead\plainheaderodd
274   \fi}
```

`\@plainheadereven` `plain` ページスタイルで使われる、ヘッダの定義です。`@plainheadereven` は、`\@plainheaderodd` `@swapheader` 無効時に偶数ページ側に、`@plainheaderodd` は、`@swapheader` 無効時に奇数ページ側に、表示されるヘッダを定義しています。

```
275 \def\@plainheadereven{\vbox{%
276   \hbox to\textwidth{%
277     \@subtitle\hfil%
278   }
279   \vskip.05\Cvs
280   \hrule}}
281 \def\@plainheaderodd{\vbox{%
282   \hbox to\textwidth{%
283     \hfil\@subtitle}
284   \vskip.05\Cvs
285   \hrule}}
```

9 文書のマークアップ

9.1 表題

`\title` これらは L^AT_EX 本体で次のように定義されています。ここではコメントアウトした形で示します。

`\author`

`\date`

```

286 % \newcommand*{\title}[1]{\gdef\@title{#1}}
287 % \newcommand*{\author}[1]{\gdef\@author{#1}}
288 % \newcommand*{\date}[1]{\gdef\@date{#1}}
289 % \date{\today}

```

`\subtitle` ヘッダに表示されるサブタイトル`\@subtitle`を定義します。初期値は`\@empty`です。

```

290 \def\subtitle#1{\gdef\@subtitle{#1}}
291 \let\@subtitle\@empty

```

`\authormark` 著者名の Prefix`\@authormark`を定義します。初期値は文`\kern1\zw{}`編集部です。

```

292 \def\authormark#1{\gdef\@authormark{#1}}
293 \newcommand{\@authormark}{文\kern1\zw{}}編集部

```

`\plainifnotempty` L^AT_EX カーネルは、様々な箇所で、ページスタイルを勝手に変更します。この問題を解決するために、「全体のページスタイルが `empty` でないならこのページのスタイルを `plain` にする」という次の命令を作ることになります。

```

294 \def\plainifnotempty{%
295   \ifx \@oddhead \@empty
296     \ifx \@oddfoot \@empty
297       \else
298         \thispagestyle{plain}%
299       \fi
300     \else
301       \thispagestyle{plain}%
302     \fi}

```

9.2 章・節

■構成要素 `\@startsection` マクロは 6 個の必須引数と、オプションとして * と 1 個のオプション引数と 1 個の必須引数をとります。

```

\@startsection{名}{レベル}{字下げ}{前アキ}{後アキ}{スタイル}
*[別見出し]{見出し}

```

それぞれの引数の意味は次の通りです。

名 ユーザレベルコマンドの名前です (例: section)。

レベル 見出しの深さを示す数値です (chapter=1, section=2, ...)。この数値が `secnumdepth` 以下のとき見出し番号を出力します。

字下げ 見出しの字下げ量です。

前アキ この値の絶対値が見出し上側の空きです。負の場合は、見出し直後の段落をインデントしません。

後アキ 正の場合は、見出しの下側の空きです。負の場合は、絶対値が見出しの右の空きです (見出しと同じ行から本文を始めます)。

スタイル 見出しの文字スタイルの設定です。

* この * 印がないと、見出し番号を付け、見出し番号のカウンタに 1 を加算します。

別見出し 目次や柱に出力する見出しです。

見出し 見出しです。

見出しの命令は通常 `\@startsection` とその最初の 6 個の引数として定義されます。

次は `\@startsection` の定義です。情報処理学会論文誌スタイルファイル (`ipsjcommon.sty`) を参考にさせていただきましたが、完全に行送りが `\baselineskip` の整数倍にならなくてもいいから前の行と重ならないようにしました。

```
303 \def\@startsection#1#2#3#4#5#6{%
304   \if@noskipsec \leavevmode \fi
305   \par
306 % 見出し上の空きを \@tempskipa にセットする
307   \@tempskipa #4\relax
308   \@afterindenttrue
309 % 見出し上の空きが負なら見出し直後の段落を字下げしない
310   \ifdim \@tempskipa <\z@
311     \@tempskipa -\@tempskipa \@afterindentfalse
312   \fi
313   \if@nobreak
314     \everypar{}%
315   \else
316     \addpenalty\@secpenalty
317 % 次の行は削除
318 %   \addvspace\@tempskipa
319 % 次の \noindent まで追加
320     \ifdim \@tempskipa >\z@
321       \null
322       \vspace*{-\baselineskip}%
323       \vskip\@tempskipa
324     \fi
325   \fi
326   \noindent
327 % 追加終わり
328   \@ifstar
329     {\@ssect{#3}{#4}{#5}{#6}}%
330     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}
```

`\@sect` と `\@xsect` は、前のアキがちょうどゼロの場合にもうまいくように、多少変えています。

```
331 \def\@sect#1#2#3#4#5#6[#7]#8{%
332   \ifnum #2>\c@secnumdepth
333     \let\@svsec\@empty
334   \else
335     \refstepcounter{#1}%
336     \protected@edef\@svsec{\@secntformat{#1}\relax}%
337   \fi
338 % 見出し後の空きを \@tempskipa にセット
339   \@tempskipa #5\relax
340 % 条件判断の順序を入れ替えました
```

```

341 \ifdim \@tempskipa<\z@
342   \def\@svsechd{%
343     #6{\hskip #3\relax
344       \@svsec #8}%
345     \csname #1mark\endcsname{#7}%
346     \addcontentsline{toc}{#1}{%
347       \ifnum #2>\c@secnumdepth \else
348         \protect\numberline{\csname the#1\endcsname}%
349       \fi
350     #7}}% 目次にフルネームを載せるなら #8
351 \else
352   \begingroup
353     \interlinepenalty \@M % 下から移動
354     #6{%
355       \@hangfrom{\hskip #3\relax\@svsec}%
356 %     \interlinepenalty \@M % 上に移動
357     #8\@@par}%
358   \endgroup
359   \csname #1mark\endcsname{#7}%
360   \addcontentsline{toc}{#1}{%
361     \ifnum #2>\c@secnumdepth \else
362       \protect\numberline{\csname the#1\endcsname}%
363     \fi
364     #7}% 目次にフルネームを載せるならここは #8
365   \fi
366   \@xsect{#5}}

```

二つ挿入した `\everyparhook` のうち後者が `\paragraph` 類の後で 2 回実行され、それ以降は前者が実行されます。

[2011-10-05 LTJ] LuaT_EX-j_a では `\everyparhook` は不要なので削除。

```

367 \def\@xsect#1{%
368 % 見出しの後ろの空きを \@tempskipa にセット
369   \@tempskipa #1\relax
370 % 条件判断の順序を変えました
371   \ifdim \@tempskipa<\z@
372     \@nbreakfalse
373     \global\@noskipsectrue
374     \everypar{%
375       \if@noskipsec
376         \global\@noskipsecfalse
377         {\setbox\z@\lastbox}%
378         \clubpenalty\@M
379       \begingroup \@svsechd \endgroup
380       \unskip
381       \@tempskipa #1\relax
382       \hskip -\@tempskipa\@inhibitglue
383     \else
384       \clubpenalty \@clubpenalty
385       \everypar{}%

```

```

386     \fi}%
387 \else
388     \par \nobreak
389     \vskip \@tempskipa
390     \@afterheading
391 \fi
392 \par % 2000-12-18
393 \ignorespaces}
394 \def\@ssect#1#2#3#4#5{%
395     \@tempskipa #3\relax
396     \ifdim \@tempskipa<\z@
397         \def\@svsechd{#4{\hskip #1\relax #5}}%
398     \else
399         \begingroup
400         #4{%
401             \@hangfrom{\hskip #1}%
402             \interlinepenalty \@M #5\@@par}%
403         \endgroup
404     \fi
405     \@xsect{#3}}

```

■柱関係の命令

`\chaptermark` `\...mark` の形の命令を初期化します (第 8 節参照)。`\chaptermark` 以外は L^AT_EX 本体で定義済みです。

```

\subsectionmark 406 \newcommand*\chaptermark[1]{}
\subsubsectionmark 407 % \newcommand*{\sectionmark}[1]{}
408 % \newcommand*{\subsectionmark}[1]{}
\paragraphmark 409 % \newcommand*{\subsubsectionmark}[1]{}
\subparagraphmark 410 % \newcommand*{\paragraphmark}[1]{}
411 % \newcommand*{\subparagraphmark}[1]{}

```

■カウンタの定義

`\c@secnumdepth` `secnumdepth` は第何レベルの見出しまで番号を付けるかを決めるカウンタです。

```
412 \setcounter{secnumdepth}{2}
```

`\c@chapter` 見出し番号のカウンタです。`\newcounter` の第 1 引数が新たに作るカウンタです。これは

`\c@section` 第 2 引数が増加するたびに 0 に戻されます。第 2 引数は定義済みのカウンタです。

```

\c@subsection 413 \newcounter{part}
\c@subsubsection 414 \newcounter{chapter}
415 \newcounter{section}
\c@paragraph 416 \newcounter{subsection}[section]
\c@subparagraph 417 \newcounter{subsubsection}[subsection]
418 \newcounter{paragraph}[subsubsection]
419 \newcounter{subparagraph}[paragraph]

```

`\thepart` カウンタの値を出力する命令 `\the` 何々 を定義します。

`\thechapter`

`\thesection`

`\thesubsection`

`\thesubsubsection`

`\theparagraph`

`\thesubparagraph`

カウンタを出力するコマンドには次のものがあります。

<code>\arabic{COUNTER}</code>	1, 2, 3, ...
<code>\roman{COUNTER}</code>	i, ii, iii, ...
<code>\Roman{COUNTER}</code>	I, II, III, ...
<code>\alph{COUNTER}</code>	a, b, c, ...
<code>\Alph{COUNTER}</code>	A, B, C, ...
<code>\kansuji{COUNTER}</code>	一, 二, 三, ...

以下ではスペース節約のため @ の付いた内部表現を多用しています。

```
420 \renewcommand{\thepart}{\@Roman\c@part}
421 \renewcommand{\thechapter}{\@arabic\c@chapter}
422 \renewcommand{\thesection}{\@arabic\c@section}
423 \renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
424 \renewcommand{\thesubsubsection}{%
425   \thesubsection.\@arabic\c@subsubsection}
426 \renewcommand{\theparagraph}{%
427   \thesubsubsection.\@arabic\c@paragraph}
428 \renewcommand{\thesubparagraph}{%
429   \theparagraph.\@arabic\c@subparagraph}
```

`\@chapapp` `\@chapapp` の初期値は `\prechaptername` (第) です。

`\@chappos` `\@chappos` の初期値は `\postchaptername` (章) です。

`\appendix` は `\@chapapp` を `\appendixname` に, `\@chappos` を空に再定義します。

[2003-03-02] `\@secapp` は外しました。

```
430 \newcommand{\@chapapp}{\prechaptername}
431 \newcommand{\@chappos}{\postchaptername}
```

■前付, 本文, 後付 本のうち章番号があるのが「本文」, それ以外が「前付」「後付」です。

`\frontmatter` ページ番号をローマ数字にし, 章番号を付けないようにします。

```
432 \newcommand\frontmatter{%
433   \clearpage
434   \@mainmatterfalse
435   \pagenumbering{roman}}
```

`\mainmatter` ページ番号を算用数字にし, 章番号を付けるようにします。

```
436 \newcommand\mainmatter{%
437   % @openright は、章を奇数起こしにするか否かのスイッチ (除去済み)
438   % word-lua.cls では常に false だが、ここでは、
439   % 条件文が何故かコメントアウトされている。
440   % \if@openright
441     \cleardoublepage
442 % \else
443 %   \clearpage
444 % \fi
445   \@mainmattertrue}
```

```
446 \pagenumbering{arabic}}
```

`\backmatter` 章番号を付けないようにします。ページ番号の付け方は変わりません。

```
447 \newcommand\backmatter{%
448 \clearpage
449 \@mainmatterfalse}
```

■部

`\part` 新しい部を始めます。

`\secdef` を使って見出しを定義しています。このマクロは二つの引数をとります。

```
\secdef{星なし}{星あり}
```

星なし * のない形の定義です。

星あり * のある形の定義です。

`\secdef` は次のようにして使います。

```
\def\chapter { ... \secdef \CMDA \CMDB }
\def\CMDA      [#1]#2{...} % \chapter[...]{...} の定義
\def\CMDB      #1{...}    % \chapter*{...} の定義
```

```
450 \newcommand\part{%
451 \clearpage
452 \thispagestyle{empty}% 欧文用標準スタイルでは plain
453 \if@twocolumn
454 \onecolumn
455 \@restonecoltrue
456 \else
457 \@restonecolfalse
458 \fi
459 \null\vfil
460 \secdef\@part\@spart}
```

`\@part` 部の見出しを出力します。`\bfseries` を `\headfont` に変えました。

```
461 \def\@part[#1]#2{%
462 \ifnum \c@secnumdepth >-2\relax
463 \refstepcounter{part}%
464 \addcontentsline{toc}{part}{%
465 \prepartname\thepart\postpartname\hspace{1\zw}#1}%
466 \else
467 \addcontentsline{toc}{part}{#1}%
468 \fi
469 \markboth{}{}%
470 {\centering
471 \interlinepenalty \@M
472 \normalfont
473 \ifnum \c@secnumdepth >-2\relax
474 \huge\headfont \prepartname\thepart\postpartname
```

```

475     \par\vskip20\p@
476     \fi
477     \Huge \headfont #2\par}%
478     \@endpart}

```

`\@spart` 番号を付けない部です。

```

479 \def\@spart#1{%
480     \centering
481     \interlinepenalty \@M
482     \normalfont
483     \Huge \headfont #1\par}%
484     \@endpart}

```

`\@endpart` `\@part` と `\@spart` の最後で実行されるマクロです。両面印刷のときは白ページを追加します。二段組のときには、二段組に戻します。

```

485 \def\@endpart{\vfil\newpage
486     \if@twoside
487     \null
488     \thispagestyle{empty}%
489     \newpage
490     \fi
491     \if@restonecol
492     \twocolumn
493     \fi}

```

■ 章

`\chapter` 章の最初のページスタイルは、全体が `empty` でなければ `plain` にします。また、`\@topnum` を 0 にして、章見出しの上に図や表が来ないようにします。

```

494 \newcommand{\chapter}{%
495     \clearpage
496     \plainifnotempty
497     \global\@topnum\z@
498     \@afterindenttrue
499     \@ifstar{\@dblarg{\@chapter}}{\@schapter}}

```

`\@chapter` 章見出しを出力します。`secnumdepth` が 0 以上かつ `\@mainmatter` が真のとき章番号を出力します。

```

500 \def\@chapter[#1]#2{%
501     \ifnum \c@secnumdepth >\m@ne
502     \if@mainmatter
503         \refstepcounter{chapter}%
504         \typeout{\@chapapp\thechapter\@chappos}%
505         \addcontentsline{toc}{chapter}%
506             {\protect\numberline
507             {\@chapapp\thechapter\@chappos}%
508             #1}%
509     \else\addcontentsline{toc}{chapter}{#1}\fi

```

```

510 \else
511   \addcontentsline{toc}{chapter}{#1}%
512 \fi
513 \chaptermark{#1}%
514 \addtocontents{lof}{\protect\addvspace{10\p@}}%
515 \addtocontents{lot}{\protect\addvspace{10\p@}}%
516 \if@twocolumn
517   \@topnewpage[\@makechapterhead{#2}]%
518 \else
519   \@makechapterhead{#2}%
520   \@afterheading
521 \fi}

```

`\@makechapterhead` 実際に章見出しを組み立てます。`\bfseries` を `\headfont` に変えました。

```

522 \def\@makechapterhead#1{%
523   {\parindent\z@
524     \raggedright
525     \reset@font\huge\gt\bfseries
526     \leavevmode
527     \ifnum \c@secnumdepth >\m@ne
528       \setlength\@tempdima{\linewidth}%
529       \if@mainmatter
530         \setbox\z@\hbox{\@chapapp\thechapter\@chappos\hskip1zw}%
531         \addtolength\@tempdima{-\wd\z@}%
532       \else
533         \setbox\z@\hbox{}%
534       \fi
535       \vtop{\hsize\@tempdima\hbox to\@tempdima{\hss\unhbox\z@\nobreak#1\hss}}%
536     \else
537       #1\relax
538     \fi}\nobreak\vskip\Cvs
539     \hbox to\textwidth{\large\hss\@authormark\hskip\Cwd\@author}%
540     \vskip.5\Cvs}

```

`\@chapter` `\chapter*{...}` コマンドの本体です。`\chaptermark` を補いました。

```

541 \def\@chapter#1{\@makeschapterhead{#1}\@afterheading}

```

`\@makeschapterhead` 番号なしの章見出しです。

```

542 \def\@makeschapterhead#1{%
543   {\parindent\z@
544     \raggedright
545     \reset@font\huge\gt\bfseries
546     \leavevmode
547     \setlength\@tempdima{\linewidth}%
548     \setbox\z@\vtop{\hbox{}\centering
549       \setlength\baselineskip{.7\baselineskip} #1}%
550     \vtop{\hsize\@tempdima
551       \centering
552       \box\z@}}

```

```

553 \par \nobreak \vskip\Cvs
554 \hbox to\textwidth{\large\hss\@authormark\hskip\Cwd\@author}%
555 \vskip.5\Cvs}

```

■下位レベルの見出し

`\section` 欧文版では `\@startsection` の第 4 引数を負にして最初の段落の字下げを禁止していますが、和文版では正にして字下げするようにしています。

段組のときはなるべく左右の段が狂わないように工夫しています。

また、 \LaTeX 向け WORD テンプレートで度々上がっていた、文の終わりと次の見出しの間は多少空けてあります。

`\section` は 3 \zw 空いています。

```

556 \newcommand{\section}{\@startsection{section}{1}{\z0}%
557     {3\zw}% 前アキ
558     {\z0}% 後アキ
559     {\reset@font\Large\gt\bfseries}}

```

`\subsection` 同上です。 `\subsection` は 2 \zw 空いています。

```

560 \newcommand{\subsection}{\@startsection{subsection}{2}{\z0}%
561     {2\zw}%
562     {\z0}%
563     {\reset@font\Large\gt\bfseries}}

```

`\subsubsection` には 1 \zw の空きを入れました。

`\subsubsection`

```

564 \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z0}%
565     {1\zw}%
566     {\z0}%
567     {\reset@font\large\gt\bfseries}}

```

`\paragraph` 見出しの後ろで改行されません。

```

568 \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z0}%
569     {\z0}%
570     {\z0}%
571     {\reset@font\normalsize\gt\bfseries}}

```

`\subparagraph` 見出しの後ろで改行されません。

```

572 \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z0}%
573     {3.25ex \@plus 1ex \@minus .2ex}%
574     {-1em}%
575     {\reset@font\normalsize\bfseries}}

```

9.3 リスト環境

ここではリスト環境について説明をしています。

リスト環境のデフォルトは次のように設定されます。

第 k レベルのリストの初期化をするのが `\@listk` です ($k = i, ii, iii, iv$)。 `\@listk` は `\leftmargin` を `\leftmargink` に設定します。

`\leftmargini` まず、`\rightmargin`, `\listparindent`, `\itemindent` をゼロにします。そして、 K 番目のレベルのリストは `\@listK` で示されるマクロが呼び出されます。ここで ' K ' は小文字のローマ数字で示されます。たとえば、3 番目のレベルのリストとして `\@listiii` が呼び出されます。 `\@listK` は `\leftmargin` を `\leftmarginK` に設定します。

```
576 \if@twocolumn
577   \setlength\leftmargini {2em}
578 \else
579   \setlength\leftmargini {2.5em}
580 \fi
```

`\leftmarginii` 次の 3 つの値は、`\labelsep` とデフォルトラベル ('(m)', 'vii.', 'M.') の幅の合計よりも大
`\leftmarginiii` きくしてあります。

```
\leftmarginiv 581 \setlength\leftmarginii {2.2em}
\leftmarginv 582 \setlength\leftmarginiii {1.87em}
\leftmarginvi 583 \setlength\leftmarginiv {1.7em}
584 \if@twocolumn
585   \setlength\leftmarginv {.5em}
586   \setlength\leftmarginvi {.5em}
587 \else
588   \setlength\leftmarginv {1em}
589   \setlength\leftmarginvi {1em}
590 \fi
```

`\labelsep` `\labelsep` はラベルとテキストの項目の間の距離です。 `\labelwidth` はラベルの幅です。

```
\labelwidth 591 \setlength \labelsep {.5em}
592 \setlength \labelwidth{\leftmargini}
593 \addtolength\labelwidth{-\labelsep}
```

`\@beginparpenalty` これらのペナルティは、リストや段落環境の前後に挿入されます。

`\@endparpenalty`
`\@itempenalty` このペナルティは、リスト項目の間に挿入されます。

```
594 \@beginparpenalty -\@lowpenalty
595 \@endparpenalty -\@lowpenalty
596 \@itempenalty -\@lowpenalty
```

`\@listi` `\@listi` は、`\leftmargin`, `\parsep`, `\topsep`, `\itemsep` などのトップレベルの定義をします。この定義は、フォントサイズコマンドによって変更されます (たとえば、`\small` の中では“小さい”リストパラメータになります)。

このため、`\normalsize` がすべてのパラメータを戻せるように、`\@listI` は `\@listi` のコピーを保存するように定義されています。

```
597 \def\@listi{\leftmargin\leftmargini
598   \parsep 2\p@ \@plus2\p@ \@minus\p@
599   \topsep 6\p@ \@plus2\p@ \@minus4\p@
600   \itemsep2\p@ \@plus2\p@ \@minus\p@}
601 \let\@listI\@listi
```

ここで、パラメータを初期化しますが、厳密には必要ありません。

602 \@listi

\@listii 下位レベルのリスト環境のパラメータの設定です。これらは保存用のバージョンを持たない
\@listiii ことと、フォントサイズコマンドによって変更されないことに注意をしてください。言い換
\@listiv えれば、このクラスは、本文サイズが\normalsize で現れるリストの入れ子についてだけ考
\@listv えています。

```
603 \def\@listii{\leftmargin\leftmarginii
604   \labelwidth\leftmarginii \advance\labelwidth-\labelsep
605   \topsep 4\p@ \@plus2\p@ \@minus\p@
606   \parsep 2\p@ \@plus\p@ \@minus\p@
607   \itemsep\parsep}
608 \def\@listiii{\leftmargin\leftmarginiii
609   \labelwidth\leftmarginiii \advance\labelwidth-\labelsep
610   \topsep 2\p@ \@plus\p@ \@minus\p@
611   \parsep\z@
612   \partopsep \p@ \@plus\z@ \@minus\p@
613   \itemsep\topsep}
614 \def\@listiv {\leftmargin\leftmarginiv
615   \labelwidth\leftmarginiv
616   \advance\labelwidth-\labelsep}
617 \def\@listv {\leftmargin\leftmarginv
618   \labelwidth\leftmarginv
619   \advance\labelwidth-\labelsep}
620 \def\@listvi {\leftmargin\leftmarginvi
621   \labelwidth\leftmarginvi
622   \advance\labelwidth-\labelsep}
```

■**enumerate 環境** enumerate 環境はカウンタ enumi, enumii, enumiii, enumiv を使います。enum*n* は第 *n* レベルの番号です。

\theenumi 出力する番号の書式を設定します。これらは L^AT_EX 本体 (ltlists.dtx 参照) で定義済み
\theenumii ですが、ここでは表し方を変えています。 \@arabic, \@alph, \@roman, \@Alph はそれぞれ
\theenumiii 算用数字, 小文字アルファベット, 小文字ローマ数字, 大文字アルファベットで番号を出
\theenumiv 力する命令です。

```
623 \renewcommand{\theenumi}{\@arabic\c@enumi}
624 \renewcommand{\theenumii}{\@alph\c@enumii}
625 \renewcommand{\theenumiii}{\@roman\c@enumiii}
626 \renewcommand{\theenumiv}{\@Alph\c@enumiv}
```

\labelenumi enumerate 環境の番号を出力する命令です。第 2 レベル以外は最後に欧文のピリオドが付
\labelenumii きますが、これは好みに応じて取り払ってください。第 2 レベルの番号のかっこは和文用に
\labelenumiii 換え、その両側に入る余分なグルーを \inhibitglue で取り除いています。

```
\labelenumiv 627 \newcommand{\labelenumi}{\theenumi.}
628 \newcommand{\labelenumii}{\inhibitglue (\theenumii) \inhibitglue}
629 \newcommand{\labelenumiii}{\theenumiii.}
630 \newcommand{\labelenumiv}{\theenumiv.}
```

`\p@enumii` `\p@enumn` は `\ref` コマンドで `enumerate` 環境の第 n レベルの項目が参照されるときに書式です。これも第 2 レベルは和文用かっこにしました。

```
\p@enumiv 631 \renewcommand{\p@enumii}{\theenumi}
632 \renewcommand{\p@enumiii}{\theenumi\inhibitglue (\theenumii) }
633 \renewcommand{\p@enumiv}{\p@enumiii\theenumiii}
```

■itemize 環境

`\labelitemi` `itemize` 環境の第 n レベルのラベルを作るコマンドです。

```
\labelitemii 634 \newcommand\labelitemi{\textbullet}
\labelitemiii 635 \newcommand\labelitemii{\normalfont\bfseries \textendash}
636 \newcommand\labelitemiii{\textasteriskcentered}
\labelitemiv 637 \newcommand\labelitemiv{\textperiodcentered}
```

■description 環境

`description` 本来の `description` 環境では、項目名が短いと、説明部分の頭がそれに引きずられて左に出てしまいます。これを解決した新しい `description` の実装です。

```
638 \newenvironment{description}{%
639 \list{}{%
640 \labelwidth=\leftmargin
641 \labelsep=1\zw
642 \advance \labelwidth by -\labelsep
643 \let \makelabel=\descriptionlabel}}{\endlist}
```

`\descriptionlabel` `description` 環境のラベルを出力するコマンドです。好みに応じて #1 の前に適当な空き（たとえば `\hspace{1\zw}`）を入れるのもいいと思います。

```
644 \newcommand*\descriptionlabel[1]{\normalfont\headfont #1\hfil}
```

■概要

`abstract` 概要（要旨、梗概）を出力する環境です。各章の初めにちょっとしたことを書くのに使います。`abstract` 環境は元は `quotation` 環境で作られていましたが、`quotation` 環境の右マージンをゼロにしたので、`list` 環境で作り直しました。

```
645 \newenvironment{abstract}{%
646 \begin{list}{}{%
647 \listparindent=1\zw
648 \itemindent=\listparindent
649 \rightmargin=0pt
650 \leftmargin=5\zw}\item[]}{\end{list}\vspace{\baselineskip}}
```

■verse 環境

`verse` 詩のための `verse` 環境です。

```
651 \newenvironment{verse}{%
652 \let \\\=\@centercr
```

```

653 \list{}{%
654   \itemsep \z@
655   \itemindent -2\zw % 元: -1.5em
656   \listparindent\itemindent
657   \rightmargin \z@
658   \advance\leftmargin 2\zw}% 元: 1.5em
659 \item\relax}{\endlist}

```

■quotation 環境

quotation 段落の頭の字下げ量を 1.5em から \parindent に変えました。また、右マージンを 0 にしました。

```

660 \newenvironment{quotation}{%
661   \list{}{%
662     \listparindent\parindent
663     \itemindent\listparindent
664     \rightmargin \z@}%
665 \item\relax}{\endlist}

```

■quote 環境

quote quote 環境は、段落がインデントされないことを除き、quotation 環境と同じです。

```

666 \newenvironment{quote}%
667 {\list{}{\rightmargin\z@}\item\relax}{\endlist}

```

■定理など ltthm.dtx 参照。たとえば次のように定義します。

```

\newtheorem{definition}{定義}
\newtheorem{axiom}{公理}
\newtheorem{theorem}{定理}

```

[2001-04-26] 定理の中はイタリック体になりましたが、これでは和文がゴシック体になってしまうので、\itshape を削除しました。

[2009-08-23] \bfseries を \headfont に直し、\labelsep を 1\zw にし、括弧を全角にしました。

```

668 \def\@begintheorem#1#2{\trivlist\labelsep=1\zw
669   \item[\hskip \labelsep{\headfont #1\ #2}]}
670 \def\@opargbegintheorem#1#2#3{\trivlist\labelsep=1\zw
671   \item[\hskip \labelsep{\headfont #1\ #2 (#3)}]}

```

■付録

\appendix 本文と付録を分離するコマンドです。

```

672 \newcommand{\appendix}{\par
673   \setcounter{chapter}{0}%
674   \setcounter{section}{0}%
675   \gdef\@chapapp{\appendixname}%

```

```

676 \gdef\chappos{}%
677 \gdef\thechapter{\@Alph\c@chapter}}

```

9.4 パラメータの設定

■array と tabular 環境

`\arraycolsep` array 環境の列間には `\arraycolsep` の 2 倍の幅の空が入ります。

```
678 \setlength\arraycolsep{5\p@}
```

`\tabcolsep` tabular 環境の列間には `\tabcolsep` の 2 倍の幅の空が入ります。

```
679 \setlength\tabcolsep{6\p@}
```

`\arrayrulewidth` array, tabular 環境内の罫線の幅です。

```
680 \setlength\arrayrulewidth{.4\p@}
```

`\doublerulesep` array, tabular 環境での二重罫線間のアキです。

```
681 \setlength\doublerulesep{2\p@}
```

■tabbing 環境

`\tabbingsep` \ ' コマンドで入るアキです。

```
682 \setlength\tabbingsep{\labelsep}
```

■minipage 環境

`\@mpfootins` minipage 環境の脚注の `\skip\@mpfootins` は通常のページの `\skip\footins` と同じ働きをします。

```
683 \skip\@mpfootins = \skip\footins
```

■framebox 環境

`\fboxsep` `\fbox`, `\framebox` で内側のテキストと枠との間の空です。

`\fboxrule` `\fbox`, `\framebox` の罫線の幅です。

```
684 \setlength\fboxsep{3\p@}
```

```
685 \setlength\fboxrule{.4\p@}
```

■equation と eqnarray 環境

`\theequation` 数式番号を出力するコマンドです。

```
686 \@addtoreset{equation}{chapter}
```

```
687 \renewcommand\theequation
```

```
688 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
```

`\jot` eqnarray の行間に余分に入るアキです。デフォルトの値をコメントアウトして示しておきます。

```
689 % \setlength\jot{3pt}
```

`\@eqnnum` 数式番号の形式です。デフォルトの値をコメントアウトして示しておきます。

`\inhibitglue (\theequation) \inhibitglue` のように和文かっこを使うことも可能です。

```
690 % \def\@eqnnum{(\theequation)}
```

`amsmath` パッケージを使う場合は `\tagform@` を次のように修正します。

```
691 % \def\tagform@#1{\maketag@@@{(\ignorespaces#1\unskip\@italiccorr)}}
```

9.5 フロート

タイプ `TYPE` のフロートオブジェクトを扱うには、次のマクロを定義します。

`\fps@TYPE` フロートを置く位置 (float placement specifier) です。

`\ftype@TYPE` フロートの番号です。2 の累乗 (1, 2, 4, ...) でなければなりません。

`\ext@TYPE` フロートの目次を出力するファイルの拡張子です。

`\fnum@TYPE` キャプション用の番号を生成するマクロです。

`\@makecaption<num><text>` キャプションを出力するマクロです。`<num>` は `\fnum@...` の生成する番号、`<text>` はキャプションのテキストです。テキストは適当な幅の `\parbox` に入ります。

■figure 環境

`\c@figure` 図番号のカウンタです。

`\thefigure` 図番号を出力するコマンドです。

```
692 \newcounter{figure}[chapter]
```

```
693 \renewcommand \thefigure
```

```
694 {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
```

`\fps@figure` `figure` のパラメータです。`\figurename` の直後に `~` が入っていましたが、ここでは外しました。

```
\ext@figure 695 \def\fps@figure{tbp}
```

```
\fnum@figure 696 \def\ftype@figure{1}
```

```
697 \def\ext@figure{lof}
```

```
698 \def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* 699 \newenvironment{figure}%
```

```
700 {\@float{figure}}%
```

```
701 {\end@float}
```

```
702 \newenvironment{figure*}%
```

```
703 {\@dblfloat{figure}}%
```

```
704 {\end@dblfloat}
```

■table 環境

`\c@table` 表番号カウンタと表番号を出力するコマンドです。アスキー版では `\thechapter.` が `\thetable` `\thechapter{}`・になっていますが、ここではオリジナルのままにしています。

```
705 \newcounter{table}[chapter]
706 \renewcommand \thetable
707     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
```

`\fps@table` `table` のパラメータです。`\tablename` の直後に ~ が入っていましたが、ここでは外しました。

```
\ext@table 708 \def\fps@table{tbp}
709 \def\ftype@table{2}
\fnun@table 710 \def\ext@table{lot}
711 \def\fnun@table{\tablename\nobreak\thetable}
```

`table` * は段抜きのフロートです。

```
table* 712 \newenvironment{table}%
713         {\@float{table}}%
714         {\end@float}
715 \newenvironment{table*}%
716         {\@dblfloat{table}}%
717         {\end@dblfloat}
```

9.6 キャプション

`\@makecaption` `\caption` コマンドにより呼び出され、実際にキャプションを出力するコマンドです。第 1 引数はフロートの番号、第 2 引数はテキストです。

`\abovecaptionskip` それぞれキャプションの前後に挿入されるスペースです。`\belowcaptionskip` が 0 になっていたので、キャプションを表の上につけた場合にキャプションと表がくっついてしまうのを直しました。

```
718 \newlength\abovecaptionskip
719 \newlength\belowcaptionskip
720 \setlength\abovecaptionskip{5\p@} % 元: 10\p@
721 \setlength\belowcaptionskip{5\p@} % 元: 0\p@
```

実際のキャプションを出力します。オリジナルと異なり、文字サイズを `\small` にし、キャプションの幅を 2cm 狭くしました。

[2003-11-05] ロジックを少し変えてみました。

```
722 \long\def\@makecaption#1#2{{\small
723   \advance\leftskip .0628\linewidth
724   \advance\rightskip .0628\linewidth
725   \vskip\abovecaptionskip
726   \sbox\@tempboxa{#1}\hskip1\zw\relax #2}%
727   \ifdim \wd\@tempboxa <\hsize \centering \fi
728   #1}\hskip1\zw\relax #2\par
729   \vskip\belowcaptionskip}}
```

10 フォントコマンド

ここでは L^AT_EX 2.09 で使われていたコマンドを定義します。これらはテキストモードと数式モードのどちらでも動作します。これらは互換性のためのもので、できるだけ `\text...` と `\math...` を使ってください。

`\mc` フォントファミリーを変更します。

```
\gt 730 \DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm 731 \DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\sf 732 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\sf 733 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt 734 \DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
```

`\bf` ボールドシリーズにします。通常のミディアムシリーズに戻るコマンドは `\mdseries` です。

```
735 \DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
```

`\it` フォントシェイプを変えるコマンドです。斜体とスモールキャップスは数式中では何もしません（警告メッセージを出力します）。通常のアップライト体に戻るコマンドは `\upshape` です。

```
736 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
737 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
738 \DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
```

`\cal` 数式モード以外では何もしません（警告を出します）。

```
\mit 739 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
740 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

11 相互参照

11.1 目次の類

`\section` コマンドは `.toc` ファイルに次のような行を出力します。

```
\contentsline{section}{タイトル}{ページ}
```

たとえば `\section` に見出し番号が付く場合、上の「タイトル」は

```
\numberline{番号}{見出し}
```

となります。この「番号」は `\thesection` コマンドで生成された見出し番号です。

`figure` 環境の `\caption` コマンドは `.lof` ファイルに次のような行を出力します。

```
\contentsline{figure}{\numberline{番号}{キャプション}}{ページ}
```

この「番号」は `\thefigure` コマンドで生成された図番号です。

table 環境も同様です。

`\contentsline{...}` は `\l@...` というコマンドを実行するので、あらかじめ `\l@chapter`, `\l@section`, `\l@figure`などを定義しておかなければなりません。これらの多くは `\@dottedtocline` コマンドを使って定義します。これは

```
\@dottedtocline{レベル}{インデント}{幅}{タイトル}{ページ}
```

という書式です。

レベル この値が `tocdepth` 以下のときだけ出力されます。`\chapter` はレベル 0, `\section` はレベル 1, 等々です。

インデント 左側の字下げ量です。

幅 「タイトル」に `\numberline` コマンドが含まれる場合、節番号が入る箱の幅です。

`\@pnumwidth` ページ番号の入る箱の幅です。

`\@tocrmarg` 右マージンです。`\@tocrmarg ≥ \@pnumwidth` とします。

`\@dotsep` 点の間隔です (単位 mu)。

`\c@tocdepth` 目次ページに出力する見出しレベルです。元は `article` で 3, その他で 2 でしたが、ここでは一つずつ減らしています。

```
741 \newcommand\@pnumwidth{1.55em}
742 \newcommand\@tocrmarg{2.55em}
743 \newcommand\@dotsep{4.5}
744 \setcounter{tocdepth}{1}
```

■目次

`\tableofcontents` 目次を生成します。

`\js@tocl@width` [2013-12-30] `\prechaptername` などから見積もった目次のラベルの長さです。(by ts)

```
745 \newdimen\js@tocl@width
746 \newcommand{\tableofcontents}{%
747   \settowidth\js@tocl@width{\headfont\prechaptername\postchaptername}%
748   \settowidth\@tempdima{\headfont\appendixname}%
749   \ifdim\js@tocl@width<\@tempdima \setlength\js@tocl@width{\@tempdima}\fi
750   \ifdim\js@tocl@width<2\zw \divide\js@tocl@width by 2 \advance\js@tocl@width 1\zw\fi
751   \if@twocolumn
752     \@restonecoltrue\onecolumn
753   \else
754     \@restonecolfalse
755   \fi
756   \chapter*{\contentsname}%
757   \@mkboth{\contentsname}{}%
758   \@starttoc{toc}%
759   \if@restonecol\twocolumn\fi
760 }
```

`\l@part` 部の目次です。

```
761 \newcommand*{\l@part}[2]{%
762   \ifnum \c@tocdepth >-2\relax
763     \addpenalty{-\@highpenalty}%
764     \addvspace{2.25em \@plus\p@}%
765     \begingroup
766       \parindent \z@
767 %     \@pnumwidth should be \@tocrmarg
768 %     \rightskip \@pnumwidth
769     \rightskip \@tocrmarg
770     \parfillskip -\rightskip
771     {\leavevmode
772       \large \headfont
773       \setlength\@lnumwidth{4\zw}%
774       #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
775     \nobreak
776     \global\@nobreaktrue
777     \everypar{\global\@nobreakfalse\everypar{}}%
778     \endgroup
779   \fi}
```

`\l@chapter` 章の目次です。 `\@lnumwidth` を `4.683\zw` に増やしました。

[2013-12-30] `\@lnumwidth` を `\js@tocl@width` から決めるようにしてみました。(by ts)

```
780 \newcommand*{\l@chapter}[2]{%
781   \ifnum \c@tocdepth >\m@ne
782     \addpenalty{-\@highpenalty}%
783     \addvspace{1.0em \@plus\p@}
784 %   \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
785     \begingroup
786       \parindent\z@
787 %     \rightskip\@pnumwidth
788     \rightskip\@tocrmarg
789     \parfillskip-\rightskip
790     \leavevmode\headfont
791     \setlength\@lnumwidth{\js@tocl@width}\advance\@lnumwidth 2.683\zw
792     \advance\leftskip\@lnumwidth \hskip-\leftskip
793     #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
794     \penalty\@highpenalty
795     \endgroup
796   \fi}
```

`\l@section` さらに下位レベルの目次項目の体裁です。あまり使ったことがありませんので、要修正かも

`\l@subsection` しれません。

`\l@subsubsection` [2013-12-30] ここの `\js@tocl@width` から決めるようにしてみました。(by ts)

```
\l@paragraph 797 \newcommand*{\l@section}{%
798               \@tempdima\js@tocl@width \advance\@tempdima -1\zw
\l@subparagraph 799               \@dottedtocline{1}{\@tempdima}{3.683\zw}}
800 \newcommand*{\l@subsection}{%
```

```

801          \@tempdima\js@tocl@width \advance\@tempdima 2.683\zw
802          \@dottedtocline{2}{\@tempdima}{3.5\zw}}
803 \newcommand*\l@subsubsection}{%
804          \@tempdima\js@tocl@width \advance\@tempdima 6.183\zw
805          \@dottedtocline{3}{\@tempdima}{4.5\zw}}
806 \newcommand*\l@paragraph}{%
807          \@tempdima\js@tocl@width \advance\@tempdima 10.683\zw
808          \@dottedtocline{4}{\@tempdima}{5.5\zw}}
809 \newcommand*\l@subparagraph}{%
810          \@tempdima\js@tocl@width \advance\@tempdima 16.183\zw
811          \@dottedtocline{5}{\@tempdima}{6.5\zw}}

```

`\numberline` 欧文版 L^AT_EX では `\numberline{...}` は幅 `\@tempdima` の箱に左詰めで出力する命令で
`\@lnumwidth` すが、アスキー版では `\@tempdima` の代わりに `\@lnumwidth` という変数で幅を決めるよう
に再定義しています。後続文字が全角か半角かでスペースが変わらないように `\hspace` を
入れておきました。

```

812 \newdimen\@lnumwidth
813 \def\numberline#1{\hbext@{\@lnumwidth{#1\hfil}\hspace{0pt}}

```

`\@dottedtocline` L^AT_EX 本体 (ltsect.dtx 参照) での定義と同じですが、`\@tempdima` を `\@lnumwidth` に
変えています。

```

814 \def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
815   \vskip \z@ \@plus.2\p@
816   {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
817     \parindent #2\relax\@afterindenttrue
818     \interlinepenalty\@M
819     \leavevmode
820     \@lnumwidth #3\relax
821     \advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
822     {#4}\nobreak
823     \leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
824       mu$}\hfill \nobreak\hbext@\@pnumwidth{%
825       \hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

826 \newcommand*\listoffigures}{%
827   \if@twocolumn\@restonecoltrue\onecolumn
828   \else\@restonecolfalse\fi
829   \chapter*{\listfigurename}%
830   \@mkboth{\listfigurename}{}%
831   \@starttoc{lof}%
832   \if@restonecol\twocolumn\fi
833 }

```

`\l@figure` 図目次の項目を出力します。

```

834 \newcommand*\l@figure}{\@dottedtocline{1}{1\zw}{3.683\zw}}

```

`\listoftables` 表目次を出力します。

```
835 \newcommand{\listoftables}{%
836   \if@twocolumn\@restonecoltrue\onecolumn
837   \else\@restonecolfalse\fi
838   \chapter*{\listtablename}%
839   \@mkboth{\listtablename}{}%
840   \@starttoc{lot}%
841   \if@restonecol\twocolumn\fi
842 }
```

`\l@table` 表目次は図目次と同じです。

```
843 \let\l@table\l@figure
```

11.2 参考文献

`\bibindent` オープンスタイルの参考文献で使うインデント幅です。元は 1.5em でした。

```
844 \newdimen\bibindent
845 \setlength\bibindent{2\zw}
```

`thebibliography` 参考文献リストを出力します。

```
846 \newenvironment{thebibliography}[1]
847 {\section*{\refname\@mkboth{\@subtitle}{\@subtitle}}%
848   \list{\@biblabel{\@arabic{c@enumiv}}}%
849     {\settowidth\labelwidth{\@biblabel{#1}}%
850       \leftmargin\labelwidth
851       \advance\leftmargin\labelsep
852       \@openbib@code
853       \usecounter{enumiv}%
854       \let\p@enumiv\@empty
855       \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
856   \sloppy
857   \clubpenalty4000
858   \@clubpenalty\clubpenalty
859   \widowpenalty4000%
860   \sfcode`\.\@m}
861 {\def\@noitemerr
862   {\@latex@warning{Empty `thebibliography' environment}}}%
863   \endlist}
```

`\newblock` `\newblock` はデフォルトでは小さなスペースを生成します。

```
864 \newcommand{\newblock}{\hspace{.11em\@plus.33em\@minus.07em}}
```

`\@openbib@code` `\@openbib@code` はデフォルトでは何もありません。この定義は `openbib` オプションによって変更されます。

```
865 \let\@openbib@code\@empty
```

`\@biblabel` `\bibitem[...]` のラベルを作ります。ltbibl.dtx の定義の半角 `[]` を全角 `【】` に変え、余

分なスペースが入らないように `\inhibitglue` ではさみました。とりあえずコメントアウトしておきますので、必要に応じて生かしてください。

```
866 % \def\biblabel#1{\inhibitglue [#1] \inhibitglue}
```

`\cite` 文献の番号を出力する部分は `ltbibl.dtx` で定義されていますが、コンマとカッコを和文
`\@cite` フォントにするには次のようにします。とりあえずコメントアウトしておきましたので、必
`\@citex` 要に応じて生かしてください。かっこの前後に入るグルーを `\inhibitglue` で取っています
 ので、オリジナル同様、Knuth~`\cite{knu}` のように半角空白で囲んでください。

```
867 % \def\@citex[#1]#2{%
868 %   \let\@citea\@empty
869 %   \@cite{\@for\@citeb:=#2\do
870 %     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
871 %     \edef\@citeb{\expandafter\@firstofone\@citeb}%
872 %     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
873 %     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}%
874 %       \G@refundefinedtrue
875 %       \@latex@warning
876 %         {Citation ``\@citeb' on page \thepage \space undefined}}%
877 %       {\hbox{\csname b@\@citeb\endcsname}}}{#1}}
878 % \def\@cite#1#2{\inhibitglue [{#1\if@tempswa , #2\fi}] \inhibitglue}
```

引用番号を上ツキの 1) のようなスタイルにするには次のようにします。`\cite` の先頭に
`\unskip` を付けて先行のスペース (~ も) を帳消しにしています。

```
879 % \DeclareRobustCommand\cite{\unskip
880 %   \@ifnextchar [{\@tempswattrue\@citex}{\@tempswafalse\@citex[]}]
881 %   \def\@cite#1#2{${\hbox{\scriptsize{#1\if@tempswa
882 %     , \inhibitglue\ #2\fi}} }$}
```

11.3 索引

`theindex` 2~3 段組の索引を作成します。最後が偶数ページのときにマージンがずれる現象を直しまし
 た (Thanks: 藤村さん)。

```
883 \newenvironment{theindex}{% 索引を 3 段組で出力する環境
884   \if@twocolumn
885     \onecolumn\@restonecolfalse
886   \else
887     \clearpage\@restonecoltrue
888   \fi
889   \columnseprule.4pt \columnsep 2\zw
890   \ifx\multicols\@undefined
891     \twocolumn[\@makeschapterhead{\indexname}]%
892     \addcontentsline{toc}{chapter}{\indexname}]%
893   \else
894     \ifdim\textwidth<\fullwidth
895       \setlength{\evensidemargin}{\oddsidemargin}
896       \setlength{\textwidth}{\fullwidth}
897       \setlength{\linewidth}{\fullwidth}
```

```

898     \begin{multicols}{3}[\chapter*{\indexname}%
899     \addcontentsline{toc}{chapter}{\indexname}}%
900     \else
901     \begin{multicols}{2}[\chapter*{\indexname}%
902     \addcontentsline{toc}{chapter}{\indexname}}%
903     \fi
904     \fi
905     \@mkboth{\indexname}{}%
906     \plainifnotempty % \thispagestyle{plain}
907     \parindent\z@
908     \parskip\z@ \@plus .3\p@\relax
909     \let\item\@idxitem
910     \raggedright
911     \footnotesize\narrowbaselines
912   }{
913     \ifx\multicols\@undefined
914     \if@restonecol\onecolumn\fi
915     \else
916     \end{multicols}
917     \fi
918     \clearpage
919   }

```

`\@idxitem` 索引項目の字下げ幅です。`\@idxitem` は `\item` の項目の字下げ幅です。

```

\subitem 920 \newcommand{\@idxitem}{\par\hangindent 40\p@}
\subsubitem 921 \newcommand{\subitem}{\@idxitem \hspace*{20\p@}}
922 \newcommand{\subsubitem}{\@idxitem \hspace*{30\p@}}

```

`\indexspace` 索引で先頭文字ごとのブロックの間に入るスペースです。

```
923 \newcommand{\indexspace}{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}
```

`\seename` 索引の `\see`, `\seealso` コマンドで出力されるものです。デフォルトはそれぞれ *see*, *see also*

`\alsoname` という英語ですが、ここではとりあえず両方とも「→」に変えました。⇒ (`\Rightarrow`) などでもいいでしょう。

```

924 \newcommand\seename{→}
925 \newcommand\alsoname{→}

```

11.4 脚注

`\footnote` 和文の句読点・閉じかっこ類の直後で用いた際に余分なアキが入るのを防ぐため、

`\footnotemark` `\inhibitglue` を入れることにします。

```

926 \let\footnotes@ve=\footnote
927 \def\footnote{\inhibitglue\footnotes@ve}
928 \let\footnotemarks@ve=\footnotemark
929 \def\footnotemark{\inhibitglue\footnotemarks@ve}

```

`\@makefnmark` 脚注番号を付ける命令です。ここでは脚注番号の前に記号 * を付けています。「注 1」の形式に

するには `\textasteriskcentered` を注`\kern0.1em`にしてください。`\@xfootnotenext`と合わせて、もし脚注番号が空なら記号も出力しないようにしてあります。

```

930 \renewcommand\@makefnmark{\hbox{\unless\ifnum\ltjgetparameter{direction}=3 $\m@th^{\@thefnmark
931     \else\hbox{\yoko$\m@th^{\@thefnmark}$}\fi}}%

```

`\thefootnote` 脚注番号に * 印が付くようにしました。ただし、番号がゼロのときは * 印も脚注番号も付きません。

```

932 \def\thefootnote{\ifnum\c@footnote>\z@\leavevmode\lower.6ex\hbox{*}\@arabic\c@footnote\fi}

```

「注 1」の形式にするには次のようにしてください。

```

933 % \def\thefootnote{\ifnum\c@footnote>\z@注\kern0.1\zw\@arabic\c@footnote\fi}

```

`\footnoterule` 本文と脚注の間の罫線です。

```

934 \renewcommand{\footnoterule}{%
935     \kern-3\p@
936     \hrule width\textwidth
937     \kern 2.6\p@}

```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```

938 \@addtoreset{footnote}{chapter}

```

`\@footnotetext` 脚注で `\verb` が使えるように改変してあります。Jeremy Gibbons, *T_EX and TUG NEWS*, Vol. 2, No. 4 (1993), p. 9

```

939 \long\def\@footnotetext{%
940     \insert\footins\bgroup
941         \normalfont\footnotesize
942         \interlinepenalty\interfootnotelinepenalty
943         \splittopskip\footnotesep
944         \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
945         \hsize\columnwidth \@parboxrestore
946         \protected@edef\@currentlabel{%
947             \csname p@footnote\endcsname\@thefnmark
948         }%
949         \color@begingroup
950         \@makefnmark{%
951             \rule{\z@\footnotesep}{\ignorespaces}%
952             \futurelet\next\fo@t}
953 \def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@t
954             \else \let\next\fo@t\fi \next}
955 \def\fo@t{\bgroup\aftergroup\@foot\let\next}
956 \def\fo@t#1{#1\@foot}
957 \def\@foot{\@finalstrut\strutbox\color@endgroup\egroup}

```

`\@makefnmark` 実際に脚注を出力する命令です。`\@makefnmark` は脚注の番号を出力する命令です。ここでは脚注が左端から一定距離に来るようにしてあります。

```

958 \newcommand\@makefnmark[1]{\parindent 1em
959     \baselineskip = 0.75\baselineskip
960     \noindent\hbox to 1.8em{\hss\@makefnmark}#1}

```

`\@xfootnotenext` 最初の `\footnotetext{...}` は番号が付きません。著者の所属などを脚注の欄に書くときに便利です。

すでに `\footnote` を使った後なら `\footnotetext[0]{...}` とすれば番号を付けない脚注になります。ただし、この場合は脚注番号がリセットされてしまうので、工夫が必要です。

12 段落の頭へのグルー挿入禁止

段落頭のかぎっこなどを見かけ 1 字半下げから全角 1 字下げに直します。

[2012-04-24 LTJ] LuaTeX-jā では JFM に段落開始時の括弧類の字下げ幅をコントロールする機能がありますが、`\item` 直後ではラベル用のボックスが段落先頭になるため、うまく働きませんでした。形を変えて復活させます。

`\item` 命令の直後です。

```
961 \protected\def\@inhibitglue{\directlua{luatexja.jfmglue.create_beginpar_node()}}
962 \def\@item[#1]{%
963   \if@noparitem
964     \@donoparitem
965   \else
966     \if@inlabel
967       \indent \par
968     \fi
969     \ifhmode
970       \unskip\unskip \par
971     \fi
972     \if@newlist
973       \if@nobreak
974         \@nbitem
975       \else
976         \addpenalty\@beginparpenalty
977         \addvspace\@topsep
978         \addvspace{-\parskip}%
979       \fi
980     \else
981       \addpenalty\@itempenalty
982       \addvspace\itemsep
983     \fi
984     \global\@inlabeltrue
985   \fi
986   \everypar{%
987     \@minipagefalse
988     \global\@newlistfalse
989     \if@inlabel
990       \global\@inlabelfalse
991     {\setbox\z@\lastbox
992     \ifvoid\z@
```



```

993     \kern-\itemindent
994     \fi}%
995     \box\@labels
996     \penalty\z@
997     \fi
998     \if@nobreak
999         \@nobreakfalse
1000     \clubpenalty \@M
1001     \else
1002         \clubpenalty \@clubpenalty
1003         \everypar{%
1004             \fi\@inhibitglue}%
1005     \if@noitemarg
1006         \@noitemargfalse
1007     \if@nmbrlist
1008         \refstepcounter\@listctr
1009     \fi
1010     \fi
1011     \sbox\@tempboxa{\makelabel{#1}}%
1012     \global\setbox\@labels\hbox{%
1013         \unhbox\@labels
1014         \hskip \itemindent
1015         \hskip -\labelwidth
1016         \hskip -\labelsep
1017         \ifdim \wd\@tempboxa >\labelwidth
1018             \box\@tempboxa
1019         \else
1020             \hbox to\labelwidth {\unhbox\@tempboxa}%
1021         \fi
1022         \hskip \labelsep}%
1023     \ignorespaces}

```

`\@gnewline` についてはちょっと複雑な心境です。もともとの $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X } 2_{\epsilon}$ は段落の頭にグルーが入る方で統一されていました。しかし `\` の直後にはグルーが入らず、不統一でした。そこで `\` の直後にもグルーを入れるように直していただいた経緯があります。しかし、ここでは逆にグルーを入れない方で統一したいので、また元に戻してしまいました。

しかし単に戻すだけでも駄目みたいなので、ここでも最後にグルーを消しておきます。

```

1024 \def\@gnewline #1{%
1025     \ifvmode
1026         \@nolnerr
1027     \else
1028         \unskip \reserved@a {\reserved@f#1}\nobreak \hfil \break \null
1029         \inhibitglue \ignorespaces
1030     \fi}

```

13 初期設定

■いろいろな語

```
\prepartname
\postpartname 1031 \newcommand{\prepartname}{第}
\prechaptername 1032 \newcommand{\postpartname}{部}
1033 \newcommand{\prechaptername}{第}
\postchaptername 1034 \newcommand{\postchaptername}{章}
\presectionname 1035 \newcommand{\presectionname}{}% 第
\postsectionname 1036 \newcommand{\postsectionname}{}% 節

\contentsname

\listfigurename 1037 \newcommand{\contentsname}{目次}
\listtablename 1038 \newcommand{\listfigurename}{図目次}
1039 \newcommand{\listtablename}{表目次}

\refname
\bibname 1040 \newcommand{\refname}{参考文献}
\indexname 1041 \newcommand{\bibname}{参考文献}
1042 \newcommand{\indexname}{索引}

\figurename
\tablename 1043 \newcommand{\figurename}{図}
1044 \newcommand{\tablename}{表}

\appendixname
\abstractname 1045 % \newcommand{\appendixname}{付録}
1046 \newcommand{\appendixname}{付録}
```

■今日の日付 \LaTeX で処理した日付を出力します。ltjarticle などと違って、標準を西暦にし、余分な空白が入らないように改良しました。和暦にするには `\和暦` と書いてください。

```
\today
1047 \newif\if 西暦 \西暦 true
1048 \def\西暦{\西暦 true}
1049 \def\和暦{\西暦 false}
1050 \newcount\heisei \heisei\year \advance\heisei-1988\relax
1051 \def\today{%
1052   \if 西暦
1053     \number\year 年
1054     \number\month 月
1055     \number\day 日
1056   \else
1057     平成\number\heisei 年
1058     \number\month 月
```

```

1059     \number\day 日
1060   \fi}

```

■ハイフネーション例外 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ のハイフネーションルールの補足です（ペンディング：
eng-lish）

```

1061 \hyphenation{ado-be post-script ghost-script phe-nom-e-no-log-i-cal man-u-script}

```

■ページ設定 ページ設定の初期化です。

```

1062 \pagestyle{plain}
1063 \pagenumbering{arabic}
1064 \if@twocolumn
1065   \twocolumn
1066   \sloppy
1067   \flushbottom
1068 \else
1069   \onecolumn
1070   \raggedbottom
1071 \fi

```

WORD の LaTeX コンパイルサーバー上でのコンパイル時にはヒラギノフォントを埋め
込みます。

```

1072 \directlua{
1073   WORD_FONT = os.getenv"WORD_FONT"
1074
1075   if WORD_FONT then
1076     tex.print(\asluastring{\PassOptionsToPackage}, "{", WORD_FONT , "} {luatexja-preset}",
1077       \asluastring{\@ifpackageloaded{luatexja-preset}{\relax}{\usepackage{luatexja-preset}}})
1078   end
1079 }

```

以上です。