

added (2017-10-04)

p_{La}T_EX 2_ε 新ドキュメントクラス

奥村晴彦, 日本語 T_EX 開発コミュニティ

2017/10/04

1 はじめに

これは L^AT_EX3 Project の `classes.dtx` と株式会社アスキーの `jclasses.dtx` に基づいてもともと奥村晴彦により作成されたものです。現在は日本語 T_EX 開発コミュニティにより GitHub で管理されています。

<https://github.com/texjporg/jsclasses>

ライセンスは modified BSD です。

[2009-02-22] 田中琢爾氏による upL^AT_EX 対応パッチを取り込みました。

ここでは次のドキュメントクラス (スタイルファイル) を作ります。

[2017-02-13] forum:2121 の議論を機に, jsreport クラスを新設しました。従来の jsbook の `report` オプションと比べると, `abstract` 環境の使い方および挙動がアスキーの jreport に近づきました。

<code><article></code>	<code>jsarticle.cls</code>	論文・レポート用
<code><book></code>	<code>jsbook.cls</code>	書籍用
<code><report></code>	<code>jsreport.cls</code>	レポート用
<code><jspf></code>	<code>jspf.cls</code>	某学会誌用
<code><kiyou></code>	<code>kiyou.cls</code>	某紀要用

L^AT_EX 2_ε あるいは p_{La}T_EX 2_ε 標準のドキュメントクラスとの違いを説明しておきます。

■JIS フォントメトリックの使用 ここでは和文 TFM (T_EX フォントメトリック) として東京書籍印刷の小林肇さんの作られた JIS フォントメトリック `jis.tfm`, `jisg.tfm` を標準で使います。従来のフォントメトリック `min10.tfm`, `goth10.tfm` の類を使うには

```
\documentclass[mingoth]{jsarticle}
```

のように `mingoth` オプションを付けます。

■サイズオプションの扱いが違う 標準のドキュメントクラスでは本文のポイント数を指定するオプションがありましたが, ポイント数は 10, 11, 12 しかなく, それぞれ別のクラスオプションファイルを読み込むようになっていました。しかも, 標準の 10 ポイント以外で

は多少フォントのバランスが崩れることがあり、あまり便利ではありませんでした。ここでは文字サイズを増すとページを小さくし、 $\mathrm{T\!E\!X}$ の `\mag` プリミティブで全体的に拡大するという手を使って、9 ポイントや 21, 25, 30, 36, 43 ポイント, 12Q, 14Q の指定を可能にしています。

以下では実際のコードに即して説明します。

`\jsc@clsname` 文書クラスの名前です。エラーメッセージ表示などで使われます。

```
(article)\def\jsc@clsname{jsarticle}
(book)\def\jsc@clsname{jsbook}
(report)\def\jsc@clsname{jsreport}
(jspf)\def\jsc@clsname{jspf}
(kiyou)\def\jsc@clsname{kiyou}
```

`\ifjsc@needsp@tch` [2016-08-22] 従来 `jsclasses` では、 $\mathrm{p\!L\!A\!T\!E\!X}$ や $\mathrm{L\!A\!T\!E\!X}$ の不都合な点に対して、クラスファイル内で独自に対策を施していました。しかし、2016 年以降、コミュニティ版 $\mathrm{p\!L\!A\!T\!E\!X}$ が次第に対策コードをカーネル内に取り込むようになりました。そこで、新しい $\mathrm{p\!L\!A\!T\!E\!X}$ カーネルと衝突しないように、日付が古い場合だけパッチをあてる場合があります。この処理に使用するフラグを定義します。

```
\newif\ifjsc@needsp@tch
\jsc@needsp@tchfalse
```

2 オプション

これらのクラスは `\documentclass{jsarticle}` あるいは `\documentclass[オプション]{jsarticle}` 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。

```
\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。

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

`\if@openright` `\chapter`, `\part` を右ページ起こしにするかどうかです。横組の書籍では真が標準で、要するに片起こし、奇数ページ起こしになります。

```
(book|report)\newif\if@openright
```

`\if@openleft` [2017-02-24] `\chapter`, `\part` を左ページ起こしにするかどうかです。

```
(book|report)\newif\if@openleft
```

`\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。

```
(book)\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。

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

以下で各オプションを宣言します。

■用紙サイズ JIS や ISO の A0 判は面積 1 m^2 ，縦横比 $1:\sqrt{2}$ の長方形の辺の長さを mm 単位に切り捨てたものです。これを基準として順に半截しては mm 単位に切り捨てたものが A1, A2, …です。

B 判は JIS と ISO で定義が異なります。JIS では B0 判の面積が 1.5 m^2 ですが，ISO では B1 判の辺の長さが A0 判と A1 判の辺の長さの幾何平均です。したがって ISO の B0 判は $1000\text{ mm} \times 1414\text{ mm}$ です。このため， \LaTeX 2_ϵ の `b5paper` は $250\text{ mm} \times 176\text{ mm}$ ですが， $\text{p}\text{\LaTeX 2}_\epsilon$ の `b5paper` は $257\text{ mm} \times 182\text{ mm}$ になっています。ここでは $\text{p}\text{\LaTeX 2}_\epsilon$ にならって JIS に従いました。

デフォルトは `a4paper` です。

`b5var` (B5 変形, $182\text{ mm} \times 230\text{ mm}$)，`a4var` (A4 変形, $210\text{ mm} \times 283\text{ mm}$) を追加しました。

```
\DeclareOption{a3paper}{%
  \setlength\paperheight {420mm}%
  \setlength\paperwidth  {297mm}}
\DeclareOption{a4paper}{%
  \setlength\paperheight {297mm}%
  \setlength\paperwidth  {210mm}}
\DeclareOption{a5paper}{%
  \setlength\paperheight {210mm}%
  \setlength\paperwidth  {148mm}}
\DeclareOption{a6paper}{%
  \setlength\paperheight {148mm}%
  \setlength\paperwidth  {105mm}}
\DeclareOption{b4paper}{%
  \setlength\paperheight {364mm}%
  \setlength\paperwidth  {257mm}}
\DeclareOption{b5paper}{%
  \setlength\paperheight {257mm}%
  \setlength\paperwidth  {182mm}}
\DeclareOption{b6paper}{%
  \setlength\paperheight {182mm}%
  \setlength\paperwidth  {128mm}}
\DeclareOption{a4j}{%
  \setlength\paperheight {297mm}%
  \setlength\paperwidth  {210mm}}
\DeclareOption{a5j}{%
  \setlength\paperheight {210mm}%
  \setlength\paperwidth  {148mm}}
\DeclareOption{b4j}{%
  \setlength\paperheight {364mm}%
  \setlength\paperwidth  {257mm}}
\DeclareOption{b5j}{%
  \setlength\paperheight {257mm}%
  \setlength\paperwidth  {182mm}}
\DeclareOption{a4var}{%
```

```

\setlength\paperheight {283mm}%
\setlength\paperwidth {210mm}%
\DeclareOption{b5var}{%
\setlength\paperheight {230mm}%
\setlength\paperwidth {182mm}%
\DeclareOption{letterpaper}{%
\setlength\paperheight {11in}%
\setlength\paperwidth {8.5in}%
\DeclareOption{legalpaper}{%
\setlength\paperheight {14in}%
\setlength\paperwidth {8.5in}%
\DeclareOption{executivepaper}{%
\setlength\paperheight {10.5in}%
\setlength\paperwidth {7.25in}%

```

■横置き 用紙の縦と横の長さを入れ換えます。

```

\newif\if@landscape
\@landscapefalse
\DeclareOption{landscape}{\@landscapetrue}

```

■slide オプション slide を新設しました。

[2016-10-08] slide オプションは article 以外では使い物にならなかったの、簡単のため article のみで使えるオプションとしました。

```

\newif\if@slide
\@slidefalse

```

■サイズオプション 10pt, 11pt, 12pt のほかに, 8pt, 9pt, 14pt, 17pt, 21pt, 25pt, 30pt, 36pt, 43pt を追加しました。これは等比数列になるように選んだものです（従来の 20pt も残しました）。\@ptsize の定義が変だったのでご迷惑をおかけしましたが、標準的なドキュメントクラスと同様にポイント数から 10 を引いたものに直しました。

[2003-03-22] 14Q オプションを追加しました。

[2003-04-18] 12Q オプションを追加しました。

[2016-07-08] \mag を使わずに各種寸法をスケールさせるためのオプション nomag を新設しました。usemag オプションの指定で従来通りの動作となります。デフォルトは usemag です。

[2016-07-24] オプティカルサイズを調整するために NFSS ヘパッチを当てるオプション nomag* を新設しました。

```

\newcommand{\@ptsize}{0}
\newif\ifjsc@mag\jsc@magtrue
\newif\ifjsc@mag@xreal\jsc@mag@xrealfalse
\def\jsc@magscale{1}
(*article)
\DeclareOption{slide}{%
\@slidetrue\def\jsc@magscale{3.583}
\renewcommand{\@ptsize}{26}

```

```

\@landscapetrue\@titlepagetrue}
</article>
\DeclareOption{8pt}{\def\jsc@magscale{0.833}\renewcommand{\@ptsize}{-2}}
\DeclareOption{9pt}{\def\jsc@magscale{0.913}\renewcommand{\@ptsize}{-1}}
\DeclareOption{10pt}{\def\jsc@magscale{1}\renewcommand{\@ptsize}{0}}
\DeclareOption{11pt}{\def\jsc@magscale{1.095}\renewcommand{\@ptsize}{1}}
\DeclareOption{12pt}{\def\jsc@magscale{1.200}\renewcommand{\@ptsize}{2}}
\DeclareOption{14pt}{\def\jsc@magscale{1.440}\renewcommand{\@ptsize}{4}}
\DeclareOption{17pt}{\def\jsc@magscale{1.728}\renewcommand{\@ptsize}{7}}
\DeclareOption{20pt}{\def\jsc@magscale{2}\renewcommand{\@ptsize}{10}}
\DeclareOption{21pt}{\def\jsc@magscale{2.074}\renewcommand{\@ptsize}{11}}
\DeclareOption{25pt}{\def\jsc@magscale{2.488}\renewcommand{\@ptsize}{15}}
\DeclareOption{30pt}{\def\jsc@magscale{2.986}\renewcommand{\@ptsize}{20}}
\DeclareOption{36pt}{\def\jsc@magscale{3.583}\renewcommand{\@ptsize}{26}}
\DeclareOption{43pt}{\def\jsc@magscale{4.300}\renewcommand{\@ptsize}{33}}
\DeclareOption{12Q}{\def\jsc@magscale{0.923}\renewcommand{\@ptsize}{1200}}
\DeclareOption{14Q}{\def\jsc@magscale{1.077}\renewcommand{\@ptsize}{1400}}
\DeclareOption{10ptj}{\def\jsc@magscale{1.085}\renewcommand{\@ptsize}{1001}}
\DeclareOption{10.5ptj}{\def\jsc@magscale{1.139}\renewcommand{\@ptsize}{1051}}
\DeclareOption{11ptj}{\def\jsc@magscale{1.194}\renewcommand{\@ptsize}{1101}}
\DeclareOption{12ptj}{\def\jsc@magscale{1.302}\renewcommand{\@ptsize}{1201}}
\DeclareOption{usemag}{\jsc@magtrue\jsc@mag@xrealfalse}
\DeclareOption{nomag}{\jsc@magfalse\jsc@mag@xrealfalse}
\DeclareOption{nomag*}{\jsc@magfalse\jsc@mag@xrealtrue}

```

■ トンボオプション トンボ (crop marks) を出力します。実際の処理は p_{La}T_EX 2_ε 本体で行います (plcore.dtx 参照)。オプション tombow で日付付きのトンボ、オプション tombo で日付なしのトンボを出力します。これらはアスキー版のままです。カウンタ \hour, \minute は p_{La}T_EX 2_ε 本体で宣言されています。

```

\hour\time \divide\hour by 60\relax
\@tempcnta\hour \multiply\@tempcnta 60\relax
\minute\time \advance\minute-\@tempcnta
\DeclareOption{tombow}{%
  \tombowtrue \tombowdatetrue
  \setlength{\@tombowwidth}{.1\p@}%
  \@bannertoken{%
    \jobname\space(\number\year-\two@digits\month-\two@digits\day
    \space\two@digits\hour:\two@digits\minute)}}%
  \maketombowbox}
\DeclareOption{tombo}{%
  \tombowtrue \tombowdatefalse
  \setlength{\@tombowwidth}{.1\p@}%
  \maketombowbox}

```

■ 面付け オプション mentuke で幅ゼロのトンボを出力します。面付けに便利です。これもアスキー版のままです。

```

\DeclareOption{mentuke}{%

```

```

\tombowtrue \tombowdatefalse
\setlength{\@tombowwidth}{\z@}%
\maketombowbox}

```

■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。

```

\DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
\DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
\DeclareOption{var twoside}{\@twosidetrue \@mparswitchfalse}

```

■二段組 `twocolumn` で二段組になります。

```

\DeclareOption{onecolumn}{\@twocolumnfalse}
\DeclareOption{twocolumn}{\@twocolumntrue}

```

■表題ページ `titlepage` で表題・概要を独立したページに出力します。

```

\DeclareOption{titlepage}{\@titlepagetrue}
\DeclareOption{notitlepage}{\@titlepagefalse}

```

■右左起こし 書籍では章は通常は奇数ページ起こしになりますが，横組ではこれを `openright` と表すことにしてあります。 `openany` で偶数ページからでも始まるようになります。

[2017-02-24] `openright` は横組では奇数ページ起こし，縦組では偶数ページ起こしを表します。ややこしいですが，これは \LaTeX の標準クラスが西欧の横組事情しか考慮せずに，奇数ページ起こしと右起こしを一緒にしてしまったせいです。縦組での奇数ページ起こしと横組での偶数ページ起こしも表現したいので，`jsclasses` では新たに `openleft` も追加しました。

```

(book | report)\DeclareOption{openright}{\@openrighttrue\@openleftfalse}
(book | report)\DeclareOption{openleft}{\@openlefttrue\@openrightfalse}
(book | report)\DeclareOption{openany}{\@openrightfalse\@openleftfalse}

```

■`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` \LaTeX 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。

```

\def\eqnarray{%
  \stepcounter{equation}%
  \def\@currentlabel{\p@equation\theequation}%
  \global\@eqnswtrue
  \m@th
  \global\@eqcnt\z@
  \tabskip\@centering
  \let\\\@eqnacr
  $$\everycr{}\halign to\displaywidth\bgroup
    \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
    &\global\@eqcnt\@ne \hfil$\displaystyle{ }{##}{ }$\hfil
    &\global\@eqcnt\tw@ $\displaystyle{##}$\hfil\tabskip\@centering

```

```

&\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
\tabskip\z@skip
\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。森本さんにしがって訂正しました。

```

\DeclareOption{leqno}{\input{leqno.clo}}
\DeclareOption{fleqn}{\input{fleqn.clo}}
% fleqn 用の eqnarray 環境の再定義
\def\eqnarray{%
  \stepcounter{equation}%
  \def\@currentlabel{\p@equation\theequation}%
  \global\@eqnswtrue\m@th
  \global\@eqcnt\z@
  \tabskip\mathindent
  \let\=\@eqnocr
  \setlength\abovedisplayskip{\topsep}%
  \ifvmode
    \addtolength\abovedisplayskip{\partopsep}%
  \fi
  \addtolength\abovedisplayskip{\parskip}%
  \setlength\belowdisplayskip{\abovedisplayskip}%
  \setlength\belowdisplayshortskip{\abovedisplayskip}%
  \setlength\abovedisplayshortskip{\abovedisplayskip}%
  $$\everycr{}\halign to\linewidth% $$
  \bgroup
  \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnse1
  &\global\@eqcnt\@ne \hfil$\displaystyle{##}$\hfil
  &\global\@eqcnt\tw@
  $\displaystyle{##}$\hfil \tabskip\@centering
  &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
  \tabskip\z@skip\cr
}

```

■文献リスト 文献リストを open 形式（著者名や書名の後に改行が入る）で出力します。これは使われることはないのでコメントアウトしてあります。

```

% \DeclareOption{openbib}{%
%   \AtEndOfPackage{%
%     \renewcommand\@openbib@code{%
%       \advance\leftmargin\bibindent
%       \itemindent -\bibindent
%       \listparindent \itemindent
%       \parsep \z@}%
%     \renewcommand\newblock{\par}}}

```

■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 数式中では 16 通りのフォントしか使えません。AMSFonTS や mathptmx パッケージを使って数式フォントをたくさん使うと “Too many math alphabets ...” というエラーが起こってしまいます。disablejfam

オプションを付ければ、明朝・ゴシックを数式用フォントとして登録するのをやめますので、数式用フォントが二つ節約できます。いずれにしても `\textmc` や `\mbox` や `amsmath` パッケージの `\text` を使えば数式中で和文フォントが使えますので、この新ドキュメントクラスでは標準で和文フォントを数式用に登録しないことにしていたのですが、従来のドキュメントクラスの仕様に合わせることにしました。

```
\DeclareOption{disablejfam}{\@enablejfamfalse}
```

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

[2016-07-13] `\ifdraft` を定義するのをやめました。

```
\DeclareOption{draft}{\setlength\overfullrule{5pt}}
```

```
\DeclareOption{final}{\setlength\overfullrule{0pt}}
```

■和文フォントメトリックの選択 ここでは新しい `jis` フォントメトリックを標準で使いますが、古い `min10`, `goth10` などを使いたいときは `mingoth` というオプションを指定します。また、`winjis` オプションで `winjis` メトリックが使えます。

```
\newif\ifwinjis
\winjisfalse
\newif\ifmingoth
\mingothfalse
\newif\ifjisfont
\jisfontfalse
\newif\if@jsc@uplatex
\@jsc@uplatexfalse
\newif\if@jsc@autodetect
\@jsc@autodetectfalse
\DeclareOption{mingoth}{\mingothtrue}
\DeclareOption{winjis}{\winjistruetrue}
\DeclareOption{jis}{\jisfonttrue}
\DeclareOption{uplatex}{\@jsc@uplatextrue\winjisfalse}
\DeclareOption{autodetect-engine}{\@jsc@autodetecttrue}
\def\jsc@JYn{\if@jsc@uplatex JY2\else JY1\fi}
\def\jsc@JTn{\if@jsc@uplatex JT2\else JT1\fi}
\def\jsc@pfx@{\if@jsc@uplatex u\else \fi}
```

■papersize スペシャルの利用 `dvips` や `dviout` で用紙設定を自動化するにはオプション `papersize` を与えます。

```
\newif\ifpapersize
\papersizefalse
\DeclareOption{papersize}{\papersizetrue}
```

■英語化 オプション `english` を新設しました。

```
\newif\if@english
\@englishfalse
\DeclareOption{english}{\@englishtrue}
```

■jsbook を jsreport もどきに オプション `report` を新設しました。

[2017-02-13] 従来は「jsreport 相当」を jsbook の `report` オプションで提供していましたが、新しく jsreport クラスも作りました。どちらでもお好きな方を使ってください。

```

\begin{book}
\newif\if@report
\@reportfalse
\DeclareOption{report}{\@reporttrue\@openrightfalse\@twosidefalse\@mparswitchfalse}
\end{book}

```

■jslogo パッケージの読み込み L^AT_EX 関連のロゴを再定義する jslogo パッケージを読み込まないオプション nojslogo を新設しました。jslogo オプションの指定で従来どおりの動作となります。デフォルトは jslogo で、すなわちパッケージを読み込みます。

```
\newif\if@jslogo \@jslogotrue
\DeclareOption{jslogo}{\@jslogotrue}
\DeclareOption{nojslogo}{\@jslogofalse}
```

■オプションの実行 デフォルトのオプションを実行し、dvi ファイルの先頭に dvips の papersize special を書き込みます。この special は dvips や最近の dviout が対応しています。multicols や url を \RequirePackage するのはやめました。

```

<article>\ExecuteOptions{a4paper,oneside,onecolumn,notitlepage,final}
<book>\ExecuteOptions{a4paper,twoside,onecolumn,titlepage,openright,final}
<report>\ExecuteOptions{a4paper,oneside,onecolumn,titlepage,openany,final}
<jspf>\ExecuteOptions{a4paper,twoside,twocolumn,notitlepage,fleqn,final}
<kiyou>\ExecuteOptions{a4paper,twoside,twocolumn,notitlepage,final}
\ProcessOptions

```

後処理

```
\if@slide
  \def\maybeblue{\@ifundefined{ver@color.sty}{\color{blue}}}
\fi
\if@landscape
  \setlength\@tempdima {\paperheight}
  \setlength\paperheight{\paperwidth}
  \setlength\paperwidth {\@tempdima}
\fi
```

■使用エンジンの検査・自動判定 ユーザが `uplatex` オプションの有無により指定したエンジンが、実際に使われているものと一致しているかを検査し、一致しない場合はエラーメッセージを表示します。

[2016-11-09] p^LA_TE_X/ u^LA_TE_X を自動判別するオプション `autodetect-engine` を新設しました。u^LA_TE_X の場合は、グローバルオプションに `uplatex` を追加することで、自動判定に応じて `otf` パッケージにも `uplatex` オプションが渡るようにします。

```
\ifnum \ifx\ucs\undefined\z@\else\ucs"3000 \fi ="3000
\if@jsc@autodetect
\ClassInfo\jsc@clsname{Autodetected engine: upLaTeX}
\@jsc@uplatrixtrue
\g@addto@macro\@classoptionslist{\uplatrix}
```

```

\fi
\if@jsc@uplatex\else
  \ClassError\jsc@clsname
  {You are running upLaTeX.\MessageBreak
   Please use pLaTeX instead, or add 'uplatex' to\MessageBreak
   the class option list}
  {\@ehc}
\@jsc@uplatextrue
\fi

```

[2016-11-11] pL^AT_EX の場合は、オプション `uplatex` が指定されていれば必ずエラーを出します。autodetect-engine が有効になっていてもエラーを出しますが、これは `otf` パッケージに `uplatex` オプションが渡ってしまうのを防ぐためです。

```

\else
  \if@jsc@uplatex
    \ClassError\jsc@clsname
    {You are running pLaTeX.\MessageBreak
     Please use upLaTeX instead, or remove 'uplatex' from\MessageBreak
     the class option list}
    {\@ehc}
  \@jsc@uplatexfalse
\fi
\if@jsc@autodetect
  \ClassInfo\jsc@clsname{Autodetected engine: pLaTeX}
  \@jsc@uplatexfalse
\fi
\fi

```

■papersize スペシャルの出力 dvips の papersize スペシャルを出力します。これで dvips や新しい dviout で出力領域が自動設定できます。トンボの付いたときの用紙サイズは無意味ですが、いわゆる「ノビ」サイズという縦横 1 インチずつ長い用紙に出力することを考えて、1 インチずつ加えました。ところが pL^AT_EX 2_ε はトンボ出力幅を両側に 1 インチとしていますので、dvips 使用時に

```
-0 -0.5in,-0.5in
```

というオプションを与えて両側 0.5 インチのトンボにするといいでしょう。

[2003-05-17] トンボをプレビューに使うことを考えて 1 インチを 2 インチにしました。

[2016-07-11] memoir クラスのマニュアルによると、トンボを含めた用紙の寸法は `\stockwidth`, `\stockheight` と呼ぶようですので、これを使うことにしました。

[2017-01-11] トンボオプションが指定されているとき「だけ」`\stockwidth`, `\stockheight` を定義するようにしました。

```

\iftombow
  \newdimen\stockwidth \newdimen\stockheight
  \setlength{\stockwidth}{\paperwidth}
  \setlength{\stockheight}{\paperheight}
  \advance \stockwidth 2in

```

```

\advance \stockheight 2in
\fi
\ifpapersize
\iftombow
\AtBeginDvi{\special{papersize=\the\stockwidth,\the\stockheight}}
\else
\AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}
\fi
\fi

```

■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。

```

(article | book | report)\if@slide\def\n@baseline{13}\else\def\n@baseline{16}\fi
(jspf)\def\n@baseline{14.554375}
(kiyou)\def\n@baseline{14.897}

```

■拡大率の設定 サイズの変更は \TeX のプリミティブ `\mag` を使って行います。9 ポイントについては行送りも若干縮めました。サイズについては全面的に見直しました。

[2008-12-26] `1000 / \mag` に相当する `\inv@mag` を定義しました。`truein` を使っていたところを `\inv@mag in` に直しましたので、`geometry` パッケージと共存できると思います。なお、新ドキュメントクラス側で 10pt 以外にする場合の注意：

- `geometry` 側でオプション `truedimen` を指定してください。
- `geometry` 側でオプション `mag` は使えません。

[2016-07-08] `\jsc@mp` および `\jsc@mmm` に、それぞれ 1pt および 1mm を拡大させた値を格納します。以降のレイアウト指定ではこちらを使います。

```

\newdimen\jsc@mp
\newdimen\jsc@mmm
\def\inv@mag{1}
\ifjsc@mag
\jsc@mp=1\p@
\jsc@mmm=1mm
\ifnum\@ptsize=-2
\mag 833
\def\inv@mag{1.20048}
\def\n@baseline{15}%
\fi
\ifnum\@ptsize=-1
\mag 913 % formerly 900
\def\inv@mag{1.09529}
\def\n@baseline{15}%
\fi
\ifnum\@ptsize=1
\mag 1095 % formerly 1100
\def\inv@mag{0.913242}

```

```

\fi
\ifnum\@ptsize=2
  \mag 1200
  \def\inv@mag{0.833333}
\fi
\ifnum\@ptsize=4
  \mag 1440
  \def\inv@mag{0.694444}
\fi
\ifnum\@ptsize=7
  \mag 1728
  \def\inv@mag{0.578704}
\fi
\ifnum\@ptsize=10
  \mag 2000
  \def\inv@mag{0.5}
\fi
\ifnum\@ptsize=11
  \mag 2074
  \def\inv@mag{0.48216}
\fi
\ifnum\@ptsize=15
  \mag 2488
  \def\inv@mag{0.401929}
\fi
\ifnum\@ptsize=20
  \mag 2986
  \def\inv@mag{0.334896}
\fi
\ifnum\@ptsize=26
  \mag 3583
  \def\inv@mag{0.279096}
\fi
\ifnum\@ptsize=33
  \mag 4300
  \def\inv@mag{0.232558}
\fi
\ifnum\@ptsize=1200
  \mag 923
  \def\inv@mag{1.0834236}
\fi
\ifnum\@ptsize=1400
  \mag 1077
  \def\inv@mag{0.928505}
\fi
\ifnum\@ptsize=1001
  \mag 1085
  \def\inv@mag{0.921659}
\fi

```

```

\ifnum\@ptsize=1051
  \mag 1139
  \def\inv@mag{0.877963}
\fi
\ifnum\@ptsize=1101
  \mag 1194
  \def\inv@mag{0.837521}
\fi
\ifnum\@ptsize=1201
  \mag 1302
  \def\inv@mag{0.768049}
\fi
\else
  \jsc@empt=\jsc@magscale\p@
  \jsc@mmm=\jsc@magscale mm
  \def\inv@mag{1}
  \ifnum\@ptsize=-2
    \def\n@baseline{15}%
  \fi
  \ifnum\@ptsize=-1
    \def\n@baseline{15}%
  \fi
\fi
\fi
<*kiyou>
\def\jsc@magscale{0.9769230}
\ifjsc@mag
  \mag 977
  \def\inv@mag{1.02354}
  \jsc@empt=1\p@
  \jsc@mmm=1mm
\else
  \jsc@empt=\jsc@magscale\p@
  \jsc@mmm=\jsc@magscale mm
  \def\inv@mag{1}
\fi
</kiyou>
\ifjsc@mag@xreal
  \RequirePackage{type1cm}
  \mathchardef\jsc@csta=259
  \def\jsc@invscale#1#2{%
    \begingroup \@tempdima=#1\relax \@tempdimb#2\p@\relax
    \@tempcnta\@tempdima \multiply\@tempcnta\@ccclvi
    \divide\@tempcnta\@tempdimb \multiply\@tempcnta\@ccclvi
    \@tempcntb\p@ \divide\@tempcntb\@tempdimb
    \advance\@tempcnta-\@tempcntb \advance\@tempcnta-\tw@
    \@tempdimb\@tempcnta\@ne
    \advance\@tempcnta\@tempcntb \advance\@tempcnta\@tempcntb
    \advance\@tempcnta\jsc@csta \@tempdimc\@tempcnta\@ne
    \@whiledim\@tempdimb<\@tempdimc\do{%

```

```

\@tempcntb\@tempdimb \advance\@tempcntb\@tempdimc
\advance\@tempcntb\@ne \divide\@tempcntb\tw@
\ifdim #2\@tempcntb>\@tempdima
\advance\@tempcntb\m@ne \@tempdimc=\@tempcntb\@ne
\else \@tempdimb=\@tempcntb\@ne \fi}%
\xdef\jsc@gtmpa{\the\@tempdimb}%
\endgroup #1=\jsc@gtmpa\relax}
\expandafter\let\csname OT1/cmr/m/n/10\endcsname\relax
\expandafter\let\csname OMX/cmex/m/n/10\endcsname\relax
\let\jsc@get@external@font\get@external@font
\def\get@external@font{%
\jsc@preadjust@extract@font
\jsc@get@external@font}
\def\jsc@fstrunc#1{%
\edef\jsc@tmpa{\strip@pt#1}%
\expandafter\jsc@fstrunc@a\jsc@tmpa.****\@nil}
\def\jsc@fstrunc@a#1.#2#3#4#5#6\@nil{%
\if#5*\else
\edef\jsc@tmpa{#1%
\ifnum#2#3>\z@ .#2\ifnum#3>\z@ #3\fi\fi}%
\fi}
\def\jsc@preadjust@extract@font{%
\let\jsc@req@size\f@size
\dimen@f@size\p@ \jsc@invscale\dimen@\jsc@magscale
\advance\dimen@.005pt\relax \jsc@fstrunc\dimen@
\let\jsc@ref@size\jsc@tmpa
\let\f@size\jsc@ref@size}
\def\execute@size@function#1{%
\let\jsc@cref@size\f@size
\let\f@size\jsc@req@size
\csname s@fct@#1\endcsname}
\let\jsc@DeclareErrorFont\DeclareErrorFont
\def\DeclareErrorFont#1#2#3#4#5{%
\@tempdimc#5\p@ \@tempdimc\jsc@magscale\@tempdimc
\edef\jsc@tmpa{{#1}{#2}{#3}{#4}{\strip@pt\@tempdimc}}
\expandafter\jsc@DeclareErrorFont\jsc@tmpa}
\def\gen@sfcnt{%
\edef\mandatory@arg{\mandatory@arg\jsc@cref@size}%
\empty@sfcnt}
\def\genb@sfcnt{%
\edef\mandatory@arg{%
\mandatory@arg\expandafter\genb@x\jsc@cref@size..@@}%
\empty@sfcnt}
\DeclareErrorFont{OT1}{cmr}{m}{n}{10}
\fi

```

[2016-11-16] latex.ltx (ltspace.dtx) で定義されている `\smallskip` の、単位 `pt` を `\jsc@mpt` に置き換えた `\jsc@smallskip` を定義します。これは `\maketitle` で用いられます。`\jsc@medskip` と `\jsc@bigskip` は必要ないのでコメントアウトしています。

```

\jsc@smallskip
\jsc@medskip      \def\jsc@smallskip{\vspace\jsc@smallskipamount}
\jsc@bigskip      %\def\jsc@medskip{\vspace\jsc@medskipamount}
                  %\def\jsc@bigskip{\vspace\jsc@bigskipamount}

\jsc@smallskipamount
\jsc@medskipamount \newskip\jsc@smallskipamount
\jsc@bigskipamount \jsc@smallskipamount=3\jsc@empt plus 1\jsc@empt minus 1\jsc@empt
                  %\newskip\jsc@medskipamount
                  %\jsc@medskipamount =6\jsc@empt plus 2\jsc@empt minus 2\jsc@empt
                  %\newskip\jsc@bigskipamount
                  %\jsc@bigskipamoun =12\jsc@empt plus 4\jsc@empt minus 4\jsc@empt

```

[2016-07-11] 新しく追加した\stockwidth, \stockheight も\mag にあわせてスケールします。

[2017-01-11] トンボオブションが指定されているとき「だけ」\stockwidth, \stockheight が定義されています。

```

\setlength\paperwidth{\inv@mag\paperwidth}%
\setlength\paperheight{\inv@mag\paperheight}%
\iftombow
  \setlength\stockwidth{\inv@mag\stockwidth}%
  \setlength\stockheight{\inv@mag\stockheight}%
\fi

```

■pagesize スペシャルの出力 [2003-05-17] dvipdfm(x) の pagesize スペシャルを出力します。

[2004-08-08] 今の dvipdfmx は dvips 用スペシャルを理解するようなので外しました。

```

% \ifpapersize
%   \setlength{\@tempdima}{\paperwidth}
%   \setlength{\@tempdimb}{\paperheight}
%   \iftombow
%     \advance \@tempdima 2truein
%     \advance \@tempdimb 2truein
%   \fi
%   \AtBeginDvi{\special{pdf: pagesize width \the\@tempdima\space height \the\@tempdimb}}
% \fi

```

3 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, T_EX 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。

さらにややこしいことに, pT_EX (アスキーが日本語化した T_EX) の公称 10 ポイントの和

文フォント (min10 など) は, 実寸 (標準の字送り量) が 9.62216pt です。これは 3.3818mm, 写研の写植機の単位では 13.527 級, PostScript の単位では 9.5862 ポイントになります。jis フォントなどもこの値を踏襲しています。

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, $13/13.527 = 0.961$ 倍すればいいことになります。

また, ここでは和文 TFM として東京書籍印刷の小林肇さんの作られた JIS フォントメトリック (jis, jisg) を標準で使うことにしますが, mingoth オプションを付けると従来のフォントメトリック (min, goth) を使うようにします。

```

<!*jspf>
\ifwinjis
  \DeclareFontShape{JY1}{mc}{m}{n}{<-> s * [0.961] winjis}{ }
  \DeclareFontShape{JY1}{gt}{m}{n}{<-> s * [0.961] winjisg}{ }
  \DeclareFontShape{JT1}{mc}{m}{n}{<-> s * [0.961] winjis-v}{ }
  \DeclareFontShape{JT1}{gt}{m}{n}{<-> s * [0.961] winjisg-v}{ }
\else
  \ifmingoth
    \DeclareFontShape{jsc@JYn}{mc}{m}{n}{<-> s * [0.961] jsc@pfx@ min10}{ }
    \DeclareFontShape{jsc@JYn}{gt}{m}{n}{<-> s * [0.961] jsc@pfx@ goth10}{ }
    \DeclareFontShape{jsc@JTn}{mc}{m}{n}{<-> s * [0.961] jsc@pfx@ tmin10}{ }
    \DeclareFontShape{jsc@JTn}{gt}{m}{n}{<-> s * [0.961] jsc@pfx@ tgoth10}{ }
  \else
    \ifjisfont
      \DeclareFontShape{jsc@JYn}{mc}{m}{n}{<-> s * [0.961] jsc@pfx@ jis}{ }
      \DeclareFontShape{jsc@JYn}{gt}{m}{n}{<-> s * [0.961] jsc@pfx@ jisg}{ }
      \DeclareFontShape{jsc@JTn}{mc}{m}{n}{<-> s * [0.961] jsc@pfx@ tmin10}{ }
      \DeclareFontShape{jsc@JTn}{gt}{m}{n}{<-> s * [0.961] jsc@pfx@ tgoth10}{ }
    \else
      \if@jsc@uplatex
        \DeclareFontShape{JY2}{mc}{m}{n}{<-> s * [0.924690] upjisr-h}{ }
        \DeclareFontShape{JY2}{gt}{m}{n}{<-> s * [0.924690] upjisg-h}{ }
        \DeclareFontShape{JT2}{mc}{m}{n}{<-> s * [0.924690] upjisr-v}{ }
        \DeclareFontShape{JT2}{gt}{m}{n}{<-> s * [0.924690] upjisg-v}{ }
      \else
        \DeclareFontShape{jsc@JYn}{mc}{m}{n}{<-> s * [0.961] jsc@pfx@ jis}{ }
        \DeclareFontShape{jsc@JYn}{gt}{m}{n}{<-> s * [0.961] jsc@pfx@ jisg}{ }
        \DeclareFontShape{jsc@JTn}{mc}{m}{n}{<-> s * [0.961] jsc@pfx@ tmin10}{ }
        \DeclareFontShape{jsc@JTn}{gt}{m}{n}{<-> s * [0.961] jsc@pfx@ tgoth10}{ }
      \fi
    \fi
  \fi
\fi
</!*jspf>

```

9.62216 ポイントの和文フォントをさらに 0.961 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。

某学会誌では, 和文フォントを PostScript の 9 ポイントにするために, $9/(9.62216 *$

72/72.27) = 0.93885 倍します。

```

(*jspf)
\ifwinjis
  \DeclareFontShape{JY1}{mc}{m}{n}{<-> s * [0.93885] winjis}{}
  \DeclareFontShape{JY1}{gt}{m}{n}{<-> s * [0.93885] winjisg}{}
  \DeclareFontShape{JT1}{mc}{m}{n}{<-> s * [0.93885] winjis-v}{}
  \DeclareFontShape{JT1}{gt}{m}{n}{<-> s * [0.93885] winjisg-v}{}
\else
  \ifmingoth
    \DeclareFontShape{\jsc@JYn}{mc}{m}{n}{<-> s * [0.93885] \jsc@pfx@ min10}{}
    \DeclareFontShape{\jsc@JYn}{gt}{m}{n}{<-> s * [0.93885] \jsc@pfx@ goth10}{}
    \DeclareFontShape{\jsc@JTn}{mc}{m}{n}{<-> s * [0.93885] \jsc@pfx@ tmin10}{}
    \DeclareFontShape{\jsc@JTn}{gt}{m}{n}{<-> s * [0.93885] \jsc@pfx@ tgoth10}{}
  \else
    \ifjisfont
      \DeclareFontShape{\jsc@JYn}{mc}{m}{n}{<-> s * [0.93885] \jsc@pfx@ jis}{}
      \DeclareFontShape{\jsc@JYn}{gt}{m}{n}{<-> s * [0.93885] \jsc@pfx@ jisg}{}
      \DeclareFontShape{\jsc@JTn}{mc}{m}{n}{<-> s * [0.93885] \jsc@pfx@ tmin10}{}
      \DeclareFontShape{\jsc@JTn}{gt}{m}{n}{<-> s * [0.93885] \jsc@pfx@ tgoth10}{}
    \else
      \if@jsc@uplatex
        \DeclareFontShape{JY2}{mc}{m}{n}{<-> s * [0.903375] upjisr-h}{}
        \DeclareFontShape{JY2}{gt}{m}{n}{<-> s * [0.903375] upjisg-h}{}
        \DeclareFontShape{JT2}{mc}{m}{n}{<-> s * [0.903375] upjisr-v}{}
        \DeclareFontShape{JT2}{gt}{m}{n}{<-> s * [0.903375] upjisg-v}{}
      \else
        \DeclareFontShape{\jsc@JYn}{mc}{m}{n}{<-> s * [0.93885] \jsc@pfx@ jis}{}
        \DeclareFontShape{\jsc@JYn}{gt}{m}{n}{<-> s * [0.93885] \jsc@pfx@ jisg}{}
        \DeclareFontShape{\jsc@JTn}{mc}{m}{n}{<-> s * [0.93885] \jsc@pfx@ tmin10}{}
        \DeclareFontShape{\jsc@JTn}{gt}{m}{n}{<-> s * [0.93885] \jsc@pfx@ tgoth10}{}
      \fi
    \fi
  \fi
\fi

```

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。

[2003-03-16] イタリック体，斜体について，和文でゴシックを当てていましたが，数学の定理環境などで多量のイタリック体を使うことがあり，ゴシックにすると黒々となってしまうという弊害がありました。amsthm を使わない場合は定理の本文が明朝になるように \newtheorem 環境を手直ししてしのいでいましたが， $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ が数学で多用されることを考えると，イタリック体に明朝体を当てたほうがいいように思えてきましたので，イタリック体・斜体に対応する和文を明朝体に変えることにしました。

[2004-11-03] \rmfamily も和文対応にしました。

```

% \DeclareFontShape{\jsc@JYn}{mc}{bx}{n}{<->ssub*gt/m/n}{ } % in \jsc@JYnmc
% \DeclareFontShape{\jsc@JYn}{gt}{bx}{n}{<->ssub*gt/m/n}{ } % in \jsc@JYngt

```

```

\DeclareFontShape{\jsc@JYn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
\DeclareFontShape{\jsc@JYn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
\DeclareFontShape{\jsc@JYn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
\DeclareFontShape{\jsc@JYn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
\DeclareFontShape{\jsc@JYn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
\DeclareFontShape{\jsc@JYn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
\DeclareFontShape{\jsc@JYn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
% \DeclareFontShape{\jsc@JTn}{mc}{bx}{n}{<->ssub*gt/m/n}{f} % in \jsc@JTnmc
% \DeclareFontShape{\jsc@JTn}{gt}{bx}{n}{<->ssub*gt/m/n}{f} % in \jsc@JTngt
\DeclareFontShape{\jsc@JTn}{mc}{m}{it}{<->ssub*mc/m/n}{f}
\DeclareFontShape{\jsc@JTn}{mc}{m}{sl}{<->ssub*mc/m/n}{f}
\DeclareFontShape{\jsc@JTn}{mc}{m}{sc}{<->ssub*mc/m/n}{f}
\DeclareFontShape{\jsc@JTn}{gt}{m}{it}{<->ssub*gt/m/n}{f}
\DeclareFontShape{\jsc@JTn}{gt}{m}{sl}{<->ssub*gt/m/n}{f}
\DeclareFontShape{\jsc@JTn}{mc}{bx}{it}{<->ssub*gt/m/n}{f}
\DeclareFontShape{\jsc@JTn}{mc}{bx}{sl}{<->ssub*gt/m/n}{f}
\DeclareRobustCommand\rmfamily
    {\not@math@alphabet\rmfamily\mathrm
     \romanfamily\rmdefault\kanjifamily\mcdefault\selectfont}
\DeclareRobustCommand\sffamily
    {\not@math@alphabet\sffamily\mathsf
     \romanfamily\sfdefault\kanjifamily\gtdefault\selectfont}
\DeclareRobustCommand\ttfamily
    {\not@math@alphabet\ttfamily\mathtt
     \romanfamily\ttdefault\kanjifamily\gtdefault\selectfont}

```

`\textmc` 次のコマンドはイタリック補正なども含めて定義されていますが、和文ではイタリック補正
`\textgt` はあまり役に立たず、欧文・和文間のグルーが入らないという副作用もありますので、単純な定義に直します。

[2016-08-26] 和欧文間の `\xkanjiskip` が入らない問題は、`plfonts.dtx v1.3i` (2000/07/13) の時点で修正されていました。逆に、`amsmath` パッケージを読み込んだ場合に、数式内の添字で文字サイズが変化するようになるはずのところが、変わらなくなっていましたので、修正しました。

[2017-09-03] Yue ZHANG さん作の `fixjfm` パッケージが `\documentclass` より前に `\RequirePackage{fixjfm}` として読み込まれていた場合には、その定義を優先するため、このクラスファイルでは再定義しません。

[2017-09-19] 2010 年の `pTeX` の修正で、イタリック補正と和欧文間の `\xkanjiskip` の衝突が起きなくなっていますから、もうここにあるような単純化は必要ありません。ただし、このクラスファイルが古い `TeX` 環境で利用される可能性も捨てきれないので、とりあえず残しておきます。

```

\ifx\DeclareFixJFMCJKTextFontCommand\undefined
\DeclareRobustCommand\textmc[1]{%
    \relax\ifmmode \expandafter\nfss@text \fi{\mcfamily #1}}
\DeclareRobustCommand\textgt[1]{%
    \relax\ifmmode \expandafter\nfss@text \fi{\gtfamily #1}}
\fi

```

新クラスでも `disablejfam` オプションを与えなければ数式内で日本語が使えるようにしました。

さらに 2005/12/01 版の LaTeX に対応した pLaTeX に対応しました (Thanks: ymt さん)。

[2010-03-14] <http://oku.edu.mie-u.ac.jp/tex/mod/forum/discuss.php?d=411> で
の山本さんのご指摘に従って修正しました。

```
\def\reDeclareMathAlphabet#1#2#3{%
  \edef\@tempa{\expandafter\@gobble\string#2}%
  \edef\@tempb{\expandafter\@gobble\string#3}%
  \edef\@tempc{\string \@expandafter\@gobbletwo\string#2}%
  \ifx\@tempc\@tempa%
    \edef\@tempa{\expandafter\@gobbletwo\string#2}%
    \edef\@tempb{\expandafter\@gobbletwo\string#3}%
  \fi
  \begingroup
    \let\protect\noexpand
    \def\@tempaa{\relax}%
    \expandafter\ifx\csname RDMAorg@\@tempa\endcsname\relax
      \edef\@tempaa{\expandafter\def\expandafter\noexpand%
        \csname RDMAorg@\@tempa\endcsname{%
          \expandafter\noexpand\csname\@tempa\endcsname}}%
    \fi
    \def\@tempbb{\relax}%
    \expandafter\ifx\csname RDMAorg@\@tempb\endcsname\relax
      \edef\@tempbb{\expandafter\def\expandafter\noexpand%
        \csname RDMAorg@\@tempb\endcsname{%
          \expandafter\noexpand\csname\@tempb\endcsname}}%
    \fi
    \edef\@tempc{\@tempaa\@tempbb}%
  \expandafter\endgroup\@tempc%
  \edef#1{\noexpand\protect\expandafter\noexpand\csname%
    \expandafter\@gobble\string#1\space\space\endcsname}%
  \expandafter\edef\csname\expandafter\@gobble\string#1\space\space\endcsname%
    {\noexpand\DualLang@mathalph@bet%
      {\expandafter\noexpand\csname RDMAorg@\@tempa\endcsname}%
      {\expandafter\noexpand\csname RDMAorg@\@tempb\endcsname}%
    }%
}
\@onlypreamble\reDeclareMathAlphabet
\def\DualLang@mathalph@bet#1#2{%
  \relax\ifmmode
    \ifx\math@bgroup\bgroup%      2e normal style      (\mathrm{...})
      \bgroup\let\DualLang@Mfontsw\DLMfontsw@standard
    \else
      \ifx\math@bgroup\relax%      2e two letter style (\rm->\mathrm)
        \let\DualLang@Mfontsw\DLMfontsw@oldstyle
      \else

```

```

\ifx\math@bgroup\@empty% 2.09 oldlfont style ({\mathrm ...})
\let\DualLang@Mfontsw\DLFontsw@oldlfont
\else%
\panic! assume 2e normal style
\bggroup\let\DualLang@Mfontsw\DLFontsw@standard
\fi
\fi
\fi
\else
\let\DualLang@Mfontsw\@firstoftwo
\fi
\DualLang@Mfontsw{#1}{#2}%
}
\def\DLFontsw@standard#1#2#3{#1{#2{#3}}\egroup}
\def\DLFontsw@oldstyle#1#2{#1\relax\@fontswitch\relax{#2}}
\def\DLFontsw@oldlfont#1#2{#1\relax#2\relax}
\if@enablejfam
\DeclareSymbolFont{mincho}{\jsc@JYn}{mc}{m}{n}
\DeclareSymbolFontAlphabet{\mathmc}{\mincho}
\SetSymbolFont{mincho}{\bold}{\jsc@JYn}{gt}{m}{n}
\jfam\symmincho
\DeclareMathAlphabet{\mathgt}{\jsc@JYn}{gt}{m}{n}
\AtBeginDocument{%
\reDeclareMathAlphabet{\mathrm}{\@mathrm}{\@mathmc}
\reDeclareMathAlphabet{\mathbf}{\@mathbf}{\@mathgt}}
\fi

```

`\textsterling` これは `\pounds` 命令で実際に呼び出される文字です。従来からの OT1 エンコーディングでは `\$` のイタリック体が `\pounds` なので `cmti` が使われていましたが、1994 年春からは `cmu` (upright italic, 直立イタリック体) に変わりました。しかし `cmu` はその性格からして実験的なものであり、`\pounds` 以外で使われるとは思えないので、ここでは `cmti` に戻してしまいます。

[2003-08-20] Computer Modern フォントを使う機会も減り、T1 エンコーディングが一般的になってきました。この定義はもうあまり意味がないので消します。

```
% \DeclareTextCommand{\textsterling}{OT1}{\itshape\char‘\$}}
```

禁則パラメータも若干修正します。

アスキーの `kinsoku.dtx` では次の三つが 5000 に設定されています。これを 10000 に再設定します。

```

\prebreakpenalty\jis"2147=10000      % 5000  '
\postbreakpenalty\jis"2148=10000     % 5000  "
\prebreakpenalty\jis"2149=10000     % 5000  "

```

「 \TeX !」「〒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。

```

\inhibitxspcode‘!=1
\inhibitxspcode‘〒=2

```

以前の版では、たとえば「ベース名. 拡張子」のように和文文字で書いたとき、ピリオドの後に四分アキが入らないようにするために

`% \xspcode' .=0`

のようにしていました。ただ、「Foo Inc. は……」のように書いたときにもスペースが入らなくなるので、ちょっとまずい修正だったかもしれません。元に戻しました。

とりあえず「ベース名. $\mbox{}$ 拡張子」と書いてください。

「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。

`\xspcode' +=3`

`\xspcode' \% =3`

これ以外に T1 エンコーディングで 80~ff の文字もすべて欧文文字ですので、両側の和文字との間にスペースが入らなければなりません。

`\xspcode' ^^80=3`

`\xspcode' ^^81=3`

`\xspcode' ^^82=3`

`\xspcode' ^^83=3`

`\xspcode' ^^84=3`

`\xspcode' ^^85=3`

`\xspcode' ^^86=3`

`\xspcode' ^^87=3`

`\xspcode' ^^88=3`

`\xspcode' ^^89=3`

`\xspcode' ^^8a=3`

`\xspcode' ^^8b=3`

`\xspcode' ^^8c=3`

`\xspcode' ^^8d=3`

`\xspcode' ^^8e=3`

`\xspcode' ^^8f=3`

`\xspcode' ^^90=3`

`\xspcode' ^^91=3`

`\xspcode' ^^92=3`

`\xspcode' ^^93=3`

`\xspcode' ^^94=3`

`\xspcode' ^^95=3`

`\xspcode' ^^96=3`

`\xspcode' ^^97=3`

`\xspcode' ^^98=3`

`\xspcode' ^^99=3`

`\xspcode' ^^9a=3`

`\xspcode' ^^9b=3`

`\xspcode' ^^9c=3`

`\xspcode' ^^9d=3`

`\xspcode' ^^9e=3`

`\xspcode' ^^9f=3`

`\xspcode' ^^a0=3`

`\xspcode' ^^a1=3`

`\xspcode' ^^a2=3`

`\xspcode' ^^a3=3`

\xspcode'^^a4=3
\xspcode'^^a5=3
\xspcode'^^a6=3
\xspcode'^^a7=3
\xspcode'^^a8=3
\xspcode'^^a9=3
\xspcode'^^aa=3
\xspcode'^^ab=3
\xspcode'^^ac=3
\xspcode'^^ad=3
\xspcode'^^ae=3
\xspcode'^^af=3
\xspcode'^^b0=3
\xspcode'^^b1=3
\xspcode'^^b2=3
\xspcode'^^b3=3
\xspcode'^^b4=3
\xspcode'^^b5=3
\xspcode'^^b6=3
\xspcode'^^b7=3
\xspcode'^^b8=3
\xspcode'^^b9=3
\xspcode'^^ba=3
\xspcode'^^bb=3
\xspcode'^^bc=3
\xspcode'^^bd=3
\xspcode'^^be=3
\xspcode'^^bf=3
\xspcode'^^c0=3
\xspcode'^^c1=3
\xspcode'^^c2=3
\xspcode'^^c3=3
\xspcode'^^c4=3
\xspcode'^^c5=3
\xspcode'^^c6=3
\xspcode'^^c7=3
\xspcode'^^c8=3
\xspcode'^^c9=3
\xspcode'^^ca=3
\xspcode'^^cb=3
\xspcode'^^cc=3
\xspcode'^^cd=3
\xspcode'^^ce=3
\xspcode'^^cf=3
\xspcode'^^d0=3
\xspcode'^^d1=3
\xspcode'^^d2=3
\xspcode'^^d3=3
\xspcode'^^d4=3

```

\xspcode'^^d5=3
\xspcode'^^d6=3
\xspcode'^^d7=3
\xspcode'^^d8=3
\xspcode'^^d9=3
\xspcode'^^da=3
\xspcode'^^db=3
\xspcode'^^dc=3
\xspcode'^^dd=3
\xspcode'^^de=3
\xspcode'^^df=3
\xspcode'^^e0=3
\xspcode'^^e1=3
\xspcode'^^e2=3
\xspcode'^^e3=3
\xspcode'^^e4=3
\xspcode'^^e5=3
\xspcode'^^e6=3
\xspcode'^^e7=3
\xspcode'^^e8=3
\xspcode'^^e9=3
\xspcode'^^ea=3
\xspcode'^^eb=3
\xspcode'^^ec=3
\xspcode'^^ed=3
\xspcode'^^ee=3
\xspcode'^^ef=3
\xspcode'^^f0=3
\xspcode'^^f1=3
\xspcode'^^f2=3
\xspcode'^^f3=3
\xspcode'^^f4=3
\xspcode'^^f5=3
\xspcode'^^f6=3
\xspcode'^^f7=3
\xspcode'^^f8=3
\xspcode'^^f9=3
\xspcode'^^fa=3
\xspcode'^^fb=3
\xspcode'^^fc=3
\xspcode'^^fd=3
\xspcode'^^fe=3
\xspcode'^^ff=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。

[2016-07-14] 2015-01-01 の \LaTeX で，auxiliary files に書き出されたときにスペースが食

われないようにする修正が入りました。これに合わせて {} を補いました。

```
\def\@{\spacefactor3000{}}
```

4 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}
```

のようにして行います。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 L^AT_EX 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、L^AT_EX 本体で定義されています。

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

`\setfontsize` ここでは `\setfontsize` の定義を少々変更して、段落の字下げ `\parindent`、和文文字間のスペース `\kanjiskip`、和文・欧文間のスペース `\xkanjiskip` を変更しています。

`\kanjiskip` は pL^AT_EX 2_ε で 0pt plus .4pt minus .5pt に設定していますが、これはそもそも文字サイズの変更に応じて変わるべきものです。それに、プラスになったりマイナスになったりするの、追い出しと追い込みの混在が生じ、統一性を欠きます。なるべく追い出しになるようにプラスの値だけにしたいところですが、ごくわずかなマイナスは許すことにしました。

`\xkanjiskip` については、四分つまり全角の 1/4 を標準として、追い出すために三分あるいは二分まで延ばすのが一般的ですが、ここでは Times や Palatino のスペースがほぼ四分であることに着目して、これに一致させています。これなら書くときにスペースを空けても空けなくても同じ出力になります。

`\parindent` については、0 (以下) でなければ全角幅 (1zw) に直します。

[2008-02-18] english オプションで `\parindent` を 1em にしました。

```
\def\setfontsize#1#2#3{%
% \@nomath#1%
\ifx\protect\@typeset@protect
\let\@currsize#1%
\fi
\fontsize{#2}{#3}\selectfont
\ifdim\parindent>\z@
\if@english
\parindent=1em
\else
```

```

\parindent=1zw
\fi
\fi
\kanjiskip=0zw plus .1zw minus .01zw
\ifdim\xkanjiskip>\z@
\if@slide \xkanjiskip=0.1em \else
\xkanjiskip=0.25em plus 0.15em minus 0.06em
\fi
\fi}

```

`\jsc@setfontsize` クラスファイルの内部では、拡大率も考慮した `\jsc@setfontsize` を `\@setfontsize` の代わりに用いることにします。

```

\def\jsc@setfontsize#1#2#3{%
\@setfontsize#1{#2\jsc@empt}{#3\jsc@empt}}

```

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

```

\emergencystretch 3zw

```

`\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` [2003-06-30] 数式に入るところで `\narrowbaselines` を実行しているので

`\widebaselines` `\abovedisplayskip` 等が初期化されてしまうという shintok さんのご指摘に対して、しっぽ愛好家さんが次の修正を教えてくださいました。

[2008-02-18] `english` オプションで最初の段落のインデントをしないようにしました。

TODO: Hasumi さん [qa:54539] のご指摘は考慮中です。

```

\newif\ifnarrowbaselines
\if@english
\narrowbaselinestrue
\fi
\def\narrowbaselines{%
\narrowbaselinestrue
\skip0=\abovedisplayskip
\skip2=\abovedisplayshortskip
\skip4=\belowdisplayskip
\skip6=\belowdisplayshortskip
\@currsize\selectfont
\abovedisplayskip=\skip0
\abovedisplayshortskip=\skip2
\belowdisplayskip=\skip4
\belowdisplayshortskip=\skip6\relax}
\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。

本文 10 ポイントのときの行送りは、欧文の標準クラスファイルでは 12 ポイント、アスキーの和文クラスファイルでは 15 ポイントになっていますが、ここでは 16 ポイントにしました。ただし `\narrowbaselines` で欧文用の 12 ポイントになります。

公称 10 ポイントの和文フォントが約 9.25 ポイント（アスキーのものの 0.961 倍）であることもあり、行送りがかなりゆったりとしたと思います。実際、 $16/9.25 \approx 1.73$ であり、和文の推奨値の一つ「二分四分」（1.75）に近づきました。

```
\renewcommand{\normalsize}{%
  \ifnarrowbaselines
    \jsc@setfontsize\normalsize\@xpt\@xiipt
  \else
    \jsc@setfontsize\normalsize\@xpt{\n@baseline}%
  \fi
```

数式の上のアキ（`\abovedisplayskip`）、短い数式の上のアキ（`\abovedisplayshortskip`）、数式の下のアキ（`\belowdisplayshortskip`）の設定です。

[2003-02-16] ちょっと変えました。

[2009-08-26] $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ Q & A 52569 から始まる議論について逡巡していましたが、結局、微調節してみることにしました。

```
\abovedisplayskip 11\jsc@empt \@plus3\jsc@empt \@minus4\jsc@empt
\abovedisplayshortskip \z@ \@plus3\jsc@empt
\belowdisplayskip 9\jsc@empt \@plus3\jsc@empt \@minus4\jsc@empt
\belowdisplayshortskip \belowdisplayskip
```

最後に、リスト環境のトップレベルのパラメータ `\@listI` を、`\@listi` にコピーしておきます。`\@listI` の設定は後で出てきます。

```
\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。

```
\normalsize
```

`\Cht` 基準となる長さの設定をします。p $\mathrm{IAT}_{\mathrm{E}}\mathrm{X}$ 2_ε カーネル（`plfonts.dtx`）で宣言されているパラメータに実際の値を設定します。たとえば `\Cwd` は `\normalfont` の全角幅（1zw）です。

`\Cwd` [2017-08-31] 基準とする文字を「全角空白」（EUC コード 0xA1A1）から「漢」（JIS コード 0x3441）へ変更しました。

`\Cvs`

`\Chs` [2017-09-19] 内部的に使った `\box0` を空にします。

```
\setbox0\hbox{\char\jis"3441}%
\setlength\Cht{\ht0}
\setlength\Cdp{\dp0}
\setlength\Cwd{\wd0}
\setlength\Cvs{\baselineskip}
\setlength\Chs{\wd0}
\setbox0=\box\voidb@x
```

`\small` `\small` も `\normalsize` と同様に設定します。行送りは、`\normalsize` が 16 ポイントなら、割合からすれば $16 \times 0.9 = 14.4$ ポイントになりますが、`\small` の使われ方を考えて、ここでは和文 13 ポイント、欧文 11 ポイントとします。また、`\topsep` と `\parsep` は、元はそれぞれ 4 ± 2 、 2 ± 1 ポイントでしたが、ここではゼロ（`\z@`）にしました。

```
\newcommand{\small}{%
  \ifnarrowbaselines
```

```

(!kiyou) \jsc@setfontsize\small\@ixpt{11}%
(kiyou) \jsc@setfontsize\small{8.8888}{11}%
\else
(!kiyou) \jsc@setfontsize\small\@ixpt{13}%
(kiyou) \jsc@setfontsize\small{8.8888}{13.2418}%
\fi
\abovedisplayskip 9\jsc@mpt \@plus3\jsc@mpt \@minus4\jsc@mpt
\abovedisplayshortskip \z@ \@plus3\jsc@mpt
\belowdisplayskip \abovedisplayskip
\belowdisplayshortskip \belowdisplayskip
\def\@listi{\leftmargin\leftmargin
\topsep \z@
\parsep \z@
\itemsep \parsep}}

```

`\footnotesize` `\footnotesize` も同様です。`\topsep` と `\parsep` は、元はそれぞれ 3 ± 1 , 2 ± 1 ポイントでしたが、ここではゼロ (`\z@`) にしました。

```

\newcommand{\footnotesize}{%
\ifnarrowbaselines
(!kiyou) \jsc@setfontsize\footnotesize\@viipt{9.5}%
(kiyou) \jsc@setfontsize\footnotesize{8.8888}{11}%
\else
(!kiyou) \jsc@setfontsize\footnotesize\@viipt{11}%
(kiyou) \jsc@setfontsize\footnotesize{8.8888}{13.2418}%
\fi
\abovedisplayskip 6\jsc@mpt \@plus2\jsc@mpt \@minus3\jsc@mpt
\abovedisplayshortskip \z@ \@plus2\jsc@mpt
\belowdisplayskip \abovedisplayskip
\belowdisplayshortskip \belowdisplayskip
\def\@listi{\leftmargin\leftmargin
\topsep \z@
\parsep \z@
\itemsep \parsep}}

```

`\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、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行が揃うようにします。

```

\large [2004-11-03] \HUGE を追加。
\Large
\huge \newcommand{\scriptsize}{\jsc@setfontsize\scriptsize\@viipt\@viipt}
\huge \newcommand{\tiny}{\jsc@setfontsize\tiny\@vpt\@vpt}
\Huge \if@twocolumn
\HUGE (!kiyou) \newcommand{\large}{\jsc@setfontsize\large\@xiipt{\n@baseline}}
(kiyou) \newcommand{\large}{\jsc@setfontsize\large{11.111}{\n@baseline}}
\else
(!kiyou) \newcommand{\large}{\jsc@setfontsize\large\@xiipt{17}}
(kiyou) \newcommand{\large}{\jsc@setfontsize\large{11.111}{17}}
\fi

```

```

<!kiyou)\newcommand{\Large}{\jsc@setfontsize\Large\@xivpt{21}}
<kiyou)\newcommand{\Large}{\jsc@setfontsize\Large{12.222}{21}}
\newcommand{\LARGE}{\jsc@setfontsize\LARGE\@xvipt{25}}
\newcommand{\huge}{\jsc@setfontsize\huge\@xxpt{28}}
\newcommand{\Huge}{\jsc@setfontsize\Huge\@xxvpt{33}}
\newcommand{\HUGE}{\jsc@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り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。

```
\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しまいました。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。『`PLATEX` 2_ε 美文書作成入門

```

% \newcommand{\headfont}{\bfseries}
\newcommand{\headfont}{\gtfamily\sffamily}
% \newcommand{\headfont}{\sffamily\fontseries{sbc}\selectfont}

```

5 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、2zw にしました。
`\columnseprule` このスペースの中央に `\columnseprule` の幅の罫線が引かれます。

```

<!kiyou)\setlength\columnsep{2zw}
<kiyou)\setlength\columnsep{28truebp}
\setlength\columnseprule{0\jsc@empt}

```

■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。
`\normallineskip`
`\lineskiplimit` `\setlength\lineskip{1\jsc@empt}`
`\normallineskiplimit` `\setlength\normallineskip{1\jsc@empt}`
`\setlength\lineskiplimit{1\jsc@empt}`
`\setlength\normallineskiplimit{1\jsc@empt}`

`\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 `\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。

```
\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、ここではゼロにしました。`\parindent` は段落の先頭の字下げ幅です。

```
\setlength\parskip{0\jsc@mppt}  
\if@slide  
  \setlength\parindent{0zw}  
\else  
  \setlength\parindent{1zw}  
\fi
```

`\@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    \@lowpenalty    51  
                  \@medpenalty   151  
                  \@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。

```
% \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。

```
% \brokenpenalty 100
```

5.1 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 でしたが、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。

[2016-08-17] 圏点やルビが一行目に来た場合に下がるのを防ぐため、`\topskip` を 10pt から 1.38zw に増やしました。`\headheight` は従来と同じ 20pt のままとします。

```
\setlength\topskip{1.38zw}%% from 10\jsc@mppt (2016-08-17)  
\if@slide  
  \setlength\headheight{0\jsc@mppt}  
\else
```

```

\setlength\headheight{20\jsc@empt}%% from 2\topskip (2016-08-17); from \topskip (2003-06-26)
\fi

```

`\footskip` `\footskip` は本文領域下端とフッタ下端との距離です。標準クラスファイルでは、book で 0.35in (約 8.89mm), book 以外で 30pt (約 10.54mm) になっていましたが, ここでは A4 判のときちょうど 1cm となるように, `\paperheight` の 0.03367 倍 (最小 `\baselineskip`) としました。書籍については, フッタは使わないことにして, ゼロにしました。

```

(*article|kiyou)
\if@slide
\setlength\footskip{0pt}
\else
\setlength\footskip{0.03367\paperheight}
\ifdim\footskip<\baselineskip
\setlength\footskip{\baselineskip}
\fi
\fi
</article|kiyou>
<jspf>\setlength\footskip{9\jsc@mmm}
(*book)
\if@report
\setlength\footskip{0.03367\paperheight}
\ifdim\footskip<\baselineskip
\setlength\footskip{\baselineskip}
\fi
\else
\setlength\footskip{0pt}
\fi
</book>
(*report)
\setlength\footskip{0.03367\paperheight}
\ifdim\footskip<\baselineskip
\setlength\footskip{\baselineskip}
\fi
</report>

```

`\headsep` `\headsep` はヘッダ下端と本文領域上端との距離です。元は book で 18pt (約 6.33mm), それ以外で 25pt (約 8.79mm) になっていました。ここでは article は `\footskip - \topskip` としました。

[2016-10-08] article の slide のとき, および book の非 report と kiyou のときに `\headsep` を減らしそこねていたのを修正しました (2016-08-17 での修正漏れ)。

```

(*article)
\if@slide
\setlength\headsep{0\jsc@empt}
\addtolength\headsep{-\topskip}%% added (2016-10-08)
\addtolength\headsep{10\jsc@empt}%% added (2016-10-08)
\else
\setlength\headsep{\footskip}

```

```

\addtolength\headsep{-\topskip}
\fi
</article>
<*book>
\if@report
\setlength\headsep{\footskip}
\addtolength\headsep{-\topskip}
\else
\setlength\headsep{6\jsc@mmm}
\addtolength\headsep{-\topskip}%% added (2016-10-08)
\addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
\fi
</book>
<*report>
\setlength\headsep{\footskip}
\addtolength\headsep{-\topskip}
</report>
<*jspf>
\setlength\headsep{9\jsc@mmm}
\addtolength\headsep{-\topskip}
</jspf>
<*kiyou>
\setlength\headheight{0\jsc@mpt}
\setlength\headsep{0\jsc@mpt}
\addtolength\headsep{-\topskip}%% added (2016-10-08)
\addtolength\headsep{10\jsc@mpt}%% added (2016-10-08)
</kiyou>

```

\maxdepth \maxdepth は本文最下行の最大の深さで、plain TeX や L^AT_EX 2.09 では 4pt に固定でした。L^AT_EX2e では \maxdepth + \topskip を本文フォントサイズの 1.5 倍にしたいのですが、\topskip は本文フォントサイズ（ここでは 10pt）に等しいので、結局 \maxdepth は \topskip の半分の値（具体的には 5pt）にします。

```
\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なります。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。

```
\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。book 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。article では

A4 縦置きで 49 文字となります。某学会誌スタイルでは 50zw (25 文字× 2 段) + 段間 8mm とします。

```

(*article)
\if@slide
  \setlength\fullwidth{0.9\paperwidth}
\else
  \setlength\fullwidth{0.76\paperwidth}
\fi
\if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
\divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
\setlength\textwidth{\fullwidth}
\end{article}
(*book)
\if@report
  \setlength\fullwidth{0.76\paperwidth}
\else
  \setlength\fullwidth{\paperwidth}
  \addtolength\fullwidth{-36\jsc@mmm}
\fi
\if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
\divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
\setlength\textwidth{\fullwidth}
\if@report \else
  \if@twocolumn \else
    \ifdim \fullwidth>40zw
      \setlength\textwidth{40zw}
    \fi
  \fi
\fi
\end{book}
(*report)
\setlength\fullwidth{0.76\paperwidth}
\if@twocolumn \@tempdima=2zw \else \@tempdima=1zw \fi
\divide\fullwidth\@tempdima \multiply\fullwidth\@tempdima
\setlength\textwidth{\fullwidth}
\end{report}
(*jspf)
\setlength\fullwidth{50zw}
\addtolength\fullwidth{8\jsc@mmm}
\setlength\textwidth{\fullwidth}
\end{jspf}
(*kiyou)
\setlength\fullwidth{48zw}
\addtolength\fullwidth{\columnsep}
\setlength\textwidth{\fullwidth}
\end{kiyou}

```

\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。

某学会誌スタイルでは 44 行にします。

[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。

[2016-08-26] `\topskip` を 10pt から 1.38zw に増やしましたので、その分 `\textheight` を増やします (2016-08-17 での修正漏れ)。

[2016-10-08] article の slide のときに `\headheight` はゼロなので、さらに修正しました (2016-08-17 での修正漏れ)。

```
(*article | book | report)
\if@slide
  \setlength{\textheight}{0.95\paperheight}
\else
  \setlength{\textheight}{0.83\paperheight}
\fi
\addtolength{\textheight}{-10\jsc@mppt}%% from -\topskip (2016-10-08); from -\headheight (2003-
\addtolength{\textheight}{-\headsep}
\addtolength{\textheight}{-\footskip}
\addtolength{\textheight}{-\topskip}
\divide\textheight\baselineskip
\multiply\textheight\baselineskip
</article | book | report>
<jspf>\setlength{\textheight}{51\baselineskip}
<kiyou>\setlength{\textheight}{47\baselineskip}
\addtolength{\textheight}{\topskip}
\addtolength{\textheight}{0.1\jsc@mppt}
<jspf>\setlength{\mathindent}{10\jsc@mmm}
```

`\flushbottom` [2016-07-18] `\textheight` に念のため 0.1 ポイント余裕を持たせているのと同様に、`\flushbottom` にも余裕を持たせます。元の \LaTeX 2_ϵ での完全な `\flushbottom` の定義は

```
\def\flushbottom{%
  \let\@textbottom\relax \let\@texttop\relax}
```

ですが、次のようにします。

```
\def\flushbottom{%
  \def\@textbottom{\vskip \z@ \@plus.1\jsc@mppt}%
  \let\@texttop\relax}
```

`\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
`\marginparpush` みどうしの最小の間隔です。

```

\setlength\marginparsep{\columnsep}
\setlength\marginparpush{\baselineskip}

```

`\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 pLATEX 2_ε (`plcore.ltx`) 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。

```

\setlength{\oddsidemargin}{\paperwidth}
\addtolength{\oddsidemargin}{-\fullwidth}
\setlength{\oddsidemargin}{.5\oddsidemargin}
\iftombow
  \addtolength{\oddsidemargin}{-1in}
\else
  \addtolength{\oddsidemargin}{-\inv@mag in}
\fi
\setlength{\evensidemargin}{\oddsidemargin}
\if@mparswitch
  \addtolength{\evensidemargin}{\fullwidth}
  \addtolength{\evensidemargin}{-\textwidth}
\fi

```

`\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 1zw の整数倍に切り捨てます。

```

\setlength\marginparwidth{\paperwidth}
\addtolength\marginparwidth{-\oddsidemargin}
\addtolength\marginparwidth{-\inv@mag in}
\addtolength\marginparwidth{-\textwidth}
\addtolength\marginparwidth{-10\jsc@mmm}
\addtolength\marginparwidth{-\marginparsep}
@tempdima=1zw
\divide\marginparwidth\@tempdima
\multiply\marginparwidth\@tempdima

```

`\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。

[2016-08-17] `\topskip` を 10pt から 1.38zw に直しましたが，`\topmargin` は従来の値から変わらないように調節しました。…のつもりでしたが，`\textheight` を増やし忘れていたので変わってしまっていました (2016-08-26 修正済み)。

```

\setlength\topmargin{\paperheight}
\addtolength\topmargin{-\textheight}
\if@slide
  \addtolength\topmargin{-\headheight}
\else
  \addtolength\topmargin{-10\jsc@empt}%% from -\topskip (2016-10-08); from -\headheight (2003-0
\fi

```

```

\addtolength\topmargin{-\headsep}
\addtolength\topmargin{-\footskip}
\setlength\topmargin{0.5\topmargin}
(kiyou)\setlength\topmargin{81truebp}
\iftombow
\addtolength\topmargin{-1in}
\else
\addtolength\topmargin{-\inv@mag in}
\fi

```

■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。

```

{\footnotesize\global\setlength\footnotesep{\baselineskip}}
\setlength\footnotesep{0.7\footnotesep}

```

`\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。

```

\setlength{\skip\footins}{16\jsc@mpt \@plus 5\jsc@mpt \@minus 2\jsc@mpt}

```

■フロート関連 フロート (図、表) 関連のパラメータは $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 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) とフロートだけのページで設定が異なります。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。

`\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。

```

\setcounter{topnumber}{9}

```

`\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。

```

\renewcommand{\topfraction}{.85}

```

`\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。

```

\setcounter{bottomnumber}{9}

```

`\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。

```

\renewcommand{\bottomfraction}{.8}

```

`\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。

```

\setcounter{totalnumber}{20}

```

<code>\textfraction</code>	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。 <code>\renewcommand{\textfraction}{.1}</code>
<code>\floatpagefraction</code>	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。 <code>\renewcommand{\floatpagefraction}{.8}</code>
<code>\c@dbltopnumber</code>	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。 <code>\setcounter{dbltopnumber}{9}</code>
<code>\dbltopfraction</code>	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。 <code>\renewcommand{\dbltopfraction}{.8}</code>
<code>\dblfloatpagefraction</code>	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。 <code>\renewcommand{\dblfloatpagefraction}{.8}</code>
<code>\floatsep</code>	<code>\floatsep</code> はページ上部・下部のフロート間の距離です。 <code>\textfloatsep</code> はページ上部・
<code>\textfloatsep</code>	下部のフロートと本文との距離です。 <code>\intextsep</code> は本文の途中に出力されるフロートと本
<code>\intextsep</code>	文との距離です。 <code>\setlength\floatsep {12\jsc@empt \@plus 2\jsc@empt \@minus 2\jsc@empt}</code> <code>\setlength\textfloatsep{20\jsc@empt \@plus 2\jsc@empt \@minus 4\jsc@empt}</code> <code>\setlength\intextsep {12\jsc@empt \@plus 2\jsc@empt \@minus 2\jsc@empt}</code>
<code>\dblfloatsep</code>	二段組のときの段抜きのフロートについての値です。
<code>\dbltextfloatsep</code>	<code>\setlength\dblfloatsep {12\jsc@empt \@plus 2\jsc@empt \@minus 2\jsc@empt}</code> <code>\setlength\dbltextfloatsep{20\jsc@empt \@plus 2\jsc@empt \@minus 4\jsc@empt}</code>
<code>\@fptop</code>	フロートだけのページに入るグルーです。 <code>\@fptop</code> はページ上部, <code>\@fpbot</code> はページ下部,
<code>\@fpsep</code>	<code>\@fpsep</code> はフロート間に入ります。
<code>\@fpbot</code>	<code>\setlength\@fptop{0\jsc@empt \@plus 1fil}</code> <code>\setlength\@fpsep{8\jsc@empt \@plus 2fil}</code> <code>\setlength\@fpbot{0\jsc@empt \@plus 1fil}</code>
<code>\@dblfpptop</code>	段抜きフロートについての値です。
<code>\@dblfpsep</code>	<code>\setlength\@dblfpptop{0\jsc@empt \@plus 1fil}</code>
<code>\@dblfpbot</code>	<code>\setlength\@dblfpsep{8\jsc@empt \@plus 2fil}</code> <code>\setlength\@dblfpbot{0\jsc@empt \@plus 1fil}</code>

6 改ページ（日本語 T_EX 開発コミュニティ版のみ）

<code>\pltx@cleartorightpage</code>	[2017-02-24] コミュニティ版 p _A T _E X の標準クラス 2017/02/15 に合わせて、同じ命令を追
<code>\pltx@cleartoleftpage</code>	加しました。
<code>\pltx@cleartooddpage</code>	
<code>\pltx@cleartoevenpage</code>	

1. `\pltx@cleartorightpage` : 右ページになるまでページを繰る命令
2. `\pltx@cleartoleftpage` : 左ページになるまでページを繰る命令
3. `\pltx@cleartooddpage` : 奇数ページになるまでページを繰る命令
4. `\pltx@cleartoevenpage` : 偶数ページになるまでページを繰る命令

となっています。

```
(*article | book | report)
\def\pltx@cleartorightpage{\clearpage\if@twoside
\ifodd\c@page
\iftdir
\hbox{}\thispagestyle{empty}\newpage
\if@twocolumn\hbox{}\newpage\fi
\fi
\else
\ifydir
\hbox{}\thispagestyle{empty}\newpage
\if@twocolumn\hbox{}\newpage\fi
\fi
\fi\fi}
\def\pltx@cleartoleftpage{\clearpage\if@twoside
\ifodd\c@page
\ifydir
\hbox{}\thispagestyle{empty}\newpage
\if@twocolumn\hbox{}\newpage\fi
\fi
\else
\iftdir
\hbox{}\thispagestyle{empty}\newpage
\if@twocolumn\hbox{}\newpage\fi
\fi
\fi\fi}
\def\pltx@cleartooddpage{\clearpage\if@twoside
\ifodd\c@page\else
\hbox{}\thispagestyle{empty}\newpage
\if@twocolumn\hbox{}\newpage\fi
\fi\fi}
\def\pltx@cleartoevenpage{\clearpage\if@twoside
\ifodd\c@page
\hbox{}\thispagestyle{empty}\newpage
\if@twocolumn\hbox{}\newpage\fi
\fi\fi}
(/article | book | report)
```

`\cleardoublepage` [2017-02-24] コミュニティ版 p^AT_EX の標準クラス 2017/02/15 に合わせて, report と book クラスの場合に `\cleardoublepage` を再定義します。

```
(*book | report)
\if@openleft
\let\cleardoublepage\pltx@cleartoleftpage
```

```

\else\if@openright
\let\cleardoublepage\ptx@cleartorightpage
\fi\fi
</book|report>

```

7 ページスタイル

ページスタイルとして、 \LaTeX 2_ε (欧文版) の標準クラスでは `empty`, `plain`, `headings`, `myheadings` があります。このうち `empty`, `plain` スタイルは \LaTeX 2_ε 本体で定義されています。

アスキーのクラスファイルでは `headnombre`, `footnombre`, `bothstyle`, `jpl@in` が追加されていますが、ここでは欧文標準のものだけにしました。

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まともに動作します。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。 \LaTeX 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。

```

% \def\ps@empty{%
%   \let\@mkboth\@gobbletwo
%   \let\@oddhead\@empty
%   \let\@oddfoot\@empty
%   \let\@evenhead\@empty
%   \let\@evenfoot\@empty}

```

`\ps@plainhead` `plainhead` はシンプルなヘッダだけのページスタイルです。
`\ps@plainfoot` `plainfoot` はシンプルなフッタだけのページスタイルです。
`\ps@plain` `plain` は `book` では `plainhead`, それ以外では `plainfoot` になります。

```

\def\ps@plainfoot{%
  \let\@mkboth\@gobbletwo
  \let\@oddhead\@empty
  \def\@oddfoot{\normalfont\hfil\thepage\hfil}%
  \let\@evenhead\@empty
  \let\@evenfoot\@oddfoot}
\def\ps@plainhead{%
  \let\@mkboth\@gobbletwo
  \let\@oddfoot\@empty
  \let\@evenfoot\@empty
  \def\@evenhead{%
    \if@mparswitch \hss \fi
    \hbox to \fullwidth{\textbf{\thepage}\hfil}%
    \if@mparswitch\else \hss \fi}%
  \def\@oddhead{%
    \hbox to \fullwidth{\hfil\textbf{\thepage}}\hss}}
<book>\if@report \let\ps@plain\ps@plainfoot \else \let\ps@plain\ps@plainhead \fi
<!book>\let\ps@plain\ps@plainfoot

```

`\ps@headings` `headings` スタイルはヘッダに見出しとページ番号を出力します。ここではヘッダにアンダーラインを引くようにしてみました。

まず `article` の場合です。

```

<*article | kiyou>
\if@twoside
  \def\ps@headings{%
    \let\@oddfoot\@empty
    \let\@evenfoot\@empty
    \def\@evenhead{\if@mparswitch \hss \fi
      \underline{\hbox to \fullwidth{\textbf{\thepage}\hfil\leftmark}}}%
    \if@mparswitch\else \hss \fi}%
    \def\@oddhead{%
      \underline{%
        \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
    \let\@mkboth\markboth
    \def\sectionmark##1{\markboth{%
      \ifnum \c@secnumdepth >\z@ \thesection \hskip1zw\fi
      ##1}{}}%
    \def\subsectionmark##1{\markright{%
      \ifnum \c@secnumdepth >\@ne \thesubsection \hskip1zw\fi
      ##1}}%
  }
\else % if not twoside
  \def\ps@headings{%
    \let\@oddfoot\@empty
    \def\@oddhead{%
      \underline{%
        \hbox to \fullwidth{{\rightmark}\hfil\textbf{\thepage}}}\hss}%
    \let\@mkboth\markboth
  }

```



```

\def\sectionmark##1{\markright{%
\ifnum \c@secnumdepth >\z@ \thesection \hskip1zw\fi
##1}}
\fi
</article|kiyou>

```

次は book および report の場合です。[2011-05-10] しっぱ愛好家さん [qa:6370] のパッチを取り込ませていただきました（北見さん [qa:55896] のご指摘ありがとうございます）。

```

< *book | report>
\newif\if@omit@number
\def\ps@headings{%
\let\@oddfoot\@empty
\let\@evenfoot\@empty
\def\@evenhead{%
\if@mparswitch \hss \fi
\underline{\hbox to \fullwidth{\autoxspacing
\textbf{\thepage}\hfil\leftmark}}}%
\if@mparswitch\else \hss \fi}%
\def\@oddhead{\underline{\hbox to \fullwidth{\autoxspacing
{\if@twoside\rightmark\else\leftmark\fi}\hfil\textbf{\thepage}}}\hss}%
\let\@mkboth\markboth
\def\chaptermark##1{\markboth{%
\ifnum \c@secnumdepth >\m@ne
(book) \if@mainmatter
\if@omit@number\else
\@chapapp\thechapter\@chappos\hskip1zw
\fi
(book) \fi
\fi
##1}{}}%
\def\sectionmark##1{\markright{%
\ifnum \c@secnumdepth >\z@ \thesection \hskip1zw\fi
##1}}}%
</book | report>

```

最後は学会誌の場合です。

```

< *jspf>
\def\ps@headings{%
\def\@oddfoot{\normalfont\hfil\thepage\hfil}
\def\@evenfoot{\normalfont\hfil\thepage\hfil}
\def\@oddhead{\normalfont\hfil \@title \hfil}
\def\@evenhead{\normalfont\hfil ブラズマ・核融合学会誌\hfil}}
</jspf>

```

`\ps@myheadings` myheadings ページスタイルではユーザが `\markboth` や `\markright` で柱を設定するため、ここでの定義は非常に簡単です。

[2004-01-17] 渡辺徹さんのパッチを適用しました。

```

\def\ps@myheadings{%
\let\@oddfoot\@empty\let\@evenfoot\@empty

```

```

\def\@evenhead{%
  \if@mparswitch \hss \fi%
  \hbox to \fullwidth{\thepage\hfil\leftmark}%
  \if@mparswitch\else \hss \fi}%
\def\@oddhead{%
  \hbox to \fullwidth{\rightmark\hfil\thepage}\hss}%
\let\@mkboth\@gobbletwo
(book | report) \let\chaptermark\@gobble
\let\sectionmark\@gobble
(!book&!report) \let\subsectionmark\@gobble
}

```

8 文書のマークアップ

8.1 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。

```

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

```

`\etitle` 某学会誌スタイルで使う英語のタイトル, 英語の著者名, キーワード, メールアドレスです。

```

\author <*jspf>
\keywords \newcommand*\etitle{[1]{\gdef\@etitle{#1}}
\newcommand*\eauthor{[1]{\gdef\@eauthor{#1}}
\newcommand*\keywords{[1]{\gdef\@keywords{#1}}
\newcommand*\email{[1]{\gdef\@authors@mail{#1}}
\newcommand*\AuthorsEmail{[1]{\gdef\@authors@mail{author's e-mail:\ #1}}
</jspf>

```

`\plainifnotempty` 従来の標準クラスでは, 文書全体のページスタイルを `empty` にしても表題のあるページだけ `plain` になってしまうことがありました。これは `\maketitle` の定義中に `\thispagestyle{plain}` が入っているためです。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します。

```

\def\plainifnotempty{%
  \ifx \@oddhead \@empty
    \ifx \@oddfoot \@empty
      \else
        \thispagestyle{plainfoot}%
      \fi
    \else
      \thispagestyle{plainhead}%
    \fi
}

```

`\maketitle` 表題を出力します。著者名を出力する部分は、欧文の標準クラスファイルでは `\large`、和文のものでは `\Large` になっていましたが、ここでは `\large` にしました。

[2016-11-16] 新設された `nomag` および `nomag*` オプションの場合をデフォルト (`usemag` 相当) に合わせるため、`\smallskip` を `\jsc@smallskip` に置き換えました。`\smallskip` のままでは `nomag(*)` の場合にスケールしなくなり、レイアウトが変わってしまいます。

```
<*article|book|report|kiyou>
\if@titlepage
  \newcommand{\maketitle}{%
    \begin{titlepage}%
      \let\footnotesize\small
      \let\footnoterule\relax
      \let\footnote\thanks
      \null\vfil
      \if@slide
        {\footnotesize \@date}%
        \begin{center}
          \mbox{} \\\[1zw]
          \large
          {\maybeblue\hrule height0\jsc@mpt depth2\jsc@mpt\relax}\par
          \jsc@smallskip
          \@title
          \jsc@smallskip
          {\maybeblue\hrule height0\jsc@mpt depth2\jsc@mpt\relax}\par
          \vfill
          {\small \@author}%
        \end{center}
      \else
        \vskip 60\jsc@mpt
        \begin{center}%
          {\LARGE \@title \par}%
          \vskip 3em%
          {\large
            \lineskip .75em
            \begin{tabular}[t]{c}%
              \@author
            \end{tabular}\par}%
          \vskip 1.5em
          {\large \@date \par}%
        \end{center}%
      \fi
    \par
    \@thanks\vfil\null
  \end{titlepage}%
  \setcounter{footnote}{0}%
  \global\let\thanks\relax
  \global\let\maketitle\relax
  \global\let\@thanks\@empty
```

```

\global\let\@author\@empty
\global\let\@date\@empty
\global\let\@title\@empty
\global\let\title\relax
\global\let\author\relax
\global\let\date\relax
\global\let\and\relax
}%
\else
\newcommand{\maketitle}{\par
\beginingroup
\renewcommand\thefootnote{\@fnsymbol\c@footnote}%
\def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
\long\def\@makefntext##1{\advance\leftskip 3zw
\parindent 1zw\noindent
\llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3zw}##1}%
\if@twocolumn
\ifnum \col@number=\@ne
\@maketitle
\else
\twocolumn[\@maketitle]%
\fi
\else
\newpage
\global\@topnum\z@ % Prevents figures from going at top of page.
\@maketitle
\fi
\plainifnotempty
\@thanks
\endgroup
\setcounter{footnote}{0}%
\global\let\thanks\relax
\global\let\maketitle\relax
\global\let\@thanks\@empty
\global\let\@author\@empty
\global\let\@date\@empty
\global\let\@title\@empty
\global\let\title\relax
\global\let\author\relax
\global\let\date\relax
\global\let\and\relax
}

```

`\@maketitle` 独立した表題ページを作らない場合の表題の出力形式です。

```

\def\@maketitle{%
\newpage\null
\vskip 2em
\begin{center}%
\let\footnote\thanks

```

```

{\LARGE \@title \par}%
\vskip 1.5em
{\large
\lineskip .5em
\begin{tabular}[t]{c}%
\@author
\end{tabular}\par}%
\vskip 1em
{\large \@date}%
\end{center}%
\par\vskip 1.5em
<article | report | kiyou> \ifvoid\@abstractbox\else\centerline{\box\@abstractbox}\vskip1.5em\fi
}
\fi
</article | book | report | kiyou>
<*jspf>
\newcommand{\maketitle}{\par
\begingroup
\renewcommand\thefootnote{\@fnsymbol\c@footnote}%
\def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
\long\def\@makefntext##1{\advance\leftskip 3zw
\parindent 1zw\noindent
\llap{\@textsuperscript{\normalfont\@thefnmark}\hskip0.3zw}##1}%
\twocolumn[\@maketitle]%
\plainifnotempty
\@thanks
\endgroup
\setcounter{footnote}{0}%
\global\let\thanks\relax
\global\let\maketitle\relax
\global\let\@thanks\@empty
\global\let\@author\@empty
\global\let\@date\@empty
% \global\let\@title\@empty % \@title は柱に使う
\global\let\title\relax
\global\let\author\relax
\global\let\date\relax
\global\let\and\relax
\ifx\authors@mail\@undefined\else{%
\def\@makefntext{\advance\leftskip 3zw \parindent -3zw}%
\footnotetext[0]{\itshape\authors@mail}%
}\fi
\global\let\authors@mail\@undefined}
\def\@maketitle{%
\newpage\null
\vskip 6em % used to be 2em
\begin{center}
\let\footnote\thanks
\ifx\@title\@undefined\else{\LARGE\headfont\@title\par}\fi

```

```

\lineskip .5em
\ifx\@author\@undefined\else
  \vskip 1em
  \begin{tabular}[t]{c}%
    \@author
  \end{tabular}\par
\fi
\ifx\@etitle\@undefined\else
  \vskip 1em
  {\large \@etitle \par}%
\fi
\ifx\@eauthor\@undefined\else
  \vskip 1em
  \begin{tabular}[t]{c}%
    \@eauthor
  \end{tabular}\par
\fi
\vskip 1em
\@date
\end{center}
\vskip 1.5em
\centerline{\box\@abstractbox}
\ifx\@keywords\@undefined\else
  \vskip 1.5em
  \centerline{\parbox{157\jsc@mmm}{\textsf{Keywords:}}\ \small\@keywords}}
\fi
\vskip 1.5em}
</jspf>

```

8.2 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。

```
\def\@startsection#1#2#3#4#5#6{%
  \if@noskipsec \leavevmode \fi
  \par
  % 見出し上の空きを \@tempskipa にセットする
  \@tempskipa #4\relax
  % \@afterindent は見出し直後の段落を字下げするかどうかを表すスイッチ
  \if@english \@afterindentfalse \else \@afterindenttrue \fi
  % 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
  \ifdim \@tempskipa <\z@
    \@tempskipa -\@tempskipa \@afterindentfalse
  \fi
  \if@nobreak
  % \everypar{\everyparhook}% これは間違い
    \everypar{}%
  \else
    \addpenalty\@secpenalty
  % 次の行は削除
  % \addvspace\@tempskipa
  % 次の \noindent まで追加
    \ifdim \@tempskipa >\z@
      \if@slide\else
        \null
        \vspace*{-\baselineskip}%
      \fi
      \vskip\@tempskipa
    \fi
  \fi
  \noindent
  % 追加終わり
  \@ifstar
    {\@ssect{#3}{#4}{#5}{#6}}%
    {\@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くいくように、多少変えてあります。`\everyparhook` も挿入しています。

```
\def\@sect#1#2#3#4#5#6[#7]#8{%
  \ifnum #2>\c@secnumdepth
```

```

\let\@svsec\@empty
\else
\refstepcounter{#1}%
\protected@edef\@svsec{\@secntformat{#1}\relax}%
\fi
% 見出し後の空きを \@tempskipa にセット
\@tempskipa #5\relax
% 条件判断の順序を入れ替えました
\ifdim \@tempskipa<\z@
\def\@svsechd{%
#6{\hskip #3\relax
\@svsec #8}%
\csname #1mark\endcsname{#7}%
\addcontentsline{toc}{#1}{%
\ifnum #2>\c@secnumdepth \else
\protect\numberline{\csname the#1\endcsname}%
\fi
#7}}% 目次にフルネームを載せるなら #8
\else
\begingroup
\interlinepenalty \@M % 下から移動
#6{%
\@hangfrom{\hskip #3\relax\@svsec}%
% \interlinepenalty \@M % 上に移動
#8\@@par}%
\endgroup
\csname #1mark\endcsname{#7}%
\addcontentsline{toc}{#1}{%
\ifnum #2>\c@secnumdepth \else
\protect\numberline{\csname the#1\endcsname}%
\fi
#7}% 目次にフルネームを載せるならここは #8
\fi
\@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。

[2016-07-28] `slide` オプションと `twocolumn` オプションを同時に指定した場合の罫線の位置を微調整しました。

```

\def\@xsect#1{%
% 見出しの後ろの空きを \@tempskipa にセット
\@tempskipa #1\relax
% 条件判断の順序を変えました
\ifdim \@tempskipa<\z@
\@nbreakfalse
\global\@noskipsecttrue
\everypar{%
\if@noskipsec

```



```

\global\@noskipsecfalse
{\setbox\z@\lastbox}%
\clubpenalty\@M
\beginingroup \@svsechd \endgroup
\unskip
\@tempskipa #1\relax
\hskip -\@tempskipa
\else
\clubpenalty \@clubpenalty
\everypar{\everyparhook}%
\fi\everyparhook}%
\else
\par \nobreak
\vskip \@tempskipa
\@afterheading
\fi
\if@slide
{\vskip\if@twocolumn-5\jsc@empt\else-6\jsc@empt\fi
\maybeblue\hrule height0\jsc@empt depth1\jsc@empt
\vskip\if@twocolumn 4\jsc@empt\else 7\jsc@empt\fi\relax}%
\fi
\par % 2000-12-18
\ignorespaces}
\def\@ssect#1#2#3#4#5{%
\@tempskipa #3\relax
\ifdim \@tempskipa<\z@
\def\@svsechd{#4{\hskip #1\relax #5}}%
\else
\beginingroup
#4{%
\@hangfrom{\hskip #1}%
\interlinepenalty \@M #5\@@par}%
\endgroup
\fi
\@xsect{#3}}

```

■柱関係の命令

`\chaptermark` `\...mark` の形の命令を初期化します (第 7 節参照)。`\chaptermark` 以外は L^AT_EX 本体で定義済みです。

```

\subsectionmark \newcommand*\chaptermark[1]{}
\subsubsectionmark % \newcommand*{\sectionmark}[1]{}
\paragraphmark % \newcommand*{\subsectionmark}[1]{}
\subparagraphmark % \newcommand*{\subsubsectionmark}[1]{}
\subparagraphmark % \newcommand*{\paragraphmark}[1]{}
\subparagraphmark % \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定めるカウンタです。

```
(!book&!report)\setcounter{secnumdepth}{3}
(book | report)\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 \newcounter{part}
\c@subsubsection (book | report)\newcounter{chapter}
\c@paragraph (book | report)\newcounter{section}[chapter]
\c@subparagraph (!book&!report)\newcounter{section}
\c@subparagraph \newcounter{subsection}[section]
\c@subparagraph \newcounter{subsubsection}[subsection]
\c@subparagraph \newcounter{paragraph}[subsubsection]
\c@subparagraph \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` カウンタを出力するコマンドには次のものがあります。

```
\thesection \arabic{COUNTER} 1, 2, 3, ...
\thesubsection \roman{COUNTER} i, ii, iii, ...
\thesubsubsection \Roman{COUNTER} I, II, III, ...
\theparagraph \alph{COUNTER} a, b, c, ...
\thesubparagraph \Alph{COUNTER} A, B, C, ...
\thesubparagraph \kansuji{COUNTER} 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。

```
\renewcommand{\thepart}{\@Roman\c@part}
(!book&!report)% \renewcommand{\thesection}{\@arabic\c@section}
(!book&!report)\renewcommand{\thesection}{\presectionname\@arabic\c@section\postsectionname}
(!book&!report)\renewcommand{\thesubsection}{\@arabic\c@section.\@arabic\c@subsection}
(*book | report)
\renewcommand{\thechapter}{\@arabic\c@chapter}
\renewcommand{\thesection}{\thechapter.\@arabic\c@section}
\renewcommand{\thesubsection}{\thesection.\@arabic\c@subsection}
(/book | report)
\renewcommand{\thesubsubsection}{%
  \thesubsection.\@arabic\c@subsubsection}
\renewcommand{\theparagraph}{%
  \thesubsubsection.\@arabic\c@paragraph}
\renewcommand{\thesubparagraph}{%
  \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。

```
(book | report)\newcommand{\@chapapp}{\prechaptername}
```

```
(book | report)\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。

[2017-03-05] `\frontmatter` と `\mainmatter` の 2 つの命令は、改丁または改ページした後で `\pagenumbering{...}` でノンブルを 1 にリセットします。長い間 `\frontmatter` は `openany` のときに単なる改ページとしていましたが、これではノンブルをリセットする際に偶奇逆転が起こる場合があります。 `openany` かどうかに関らず奇数ページまで繰るように修正することで、問題を解消しました。実は、 \LaTeX の標準クラスでは 1998 年に修正されていた問題です (コミュニティ版 \LaTeX の標準クラス 2017/03/05 も参照)。

```
(*book)
\newcommand\frontmatter{%
  \pltx@cleartooddpage
  \@mainmatterfalse
  \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。

```
\newcommand\mainmatter{%
  \pltx@cleartooddpage
  \@mainmattertrue
  \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。

```
\newcommand\backmatter{%
  \if@openleft
    \cleardoublepage
  \else\if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi\fi
  \@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

```

まず book と report のクラス以外です。

```

(*!book&!report)
\newcommand\part{%
  \if@noskipsec \leavevmode \fi
  \par
  \addvspace{4ex}%
  \if@english \@afterindentfalse \else \@afterindenttrue \fi
  \secdef\@part\@spart}

```

book および report クラスの場合は、少し複雑です。

```

(*!book|report)
\newcommand\part{%
  \if@openleft
    \cleardoublepage
  \else\if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi\fi
  \thispagestyle{empty}% 欧文用標準スタイルでは plain
  \if@twocolumn
    \onecolumn
    \@restonecoltrue
  \else
    \@restonecolfalse
  \fi
  \null\vfil
  \secdef\@part\@spart}

```

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

book および report クラス以外では `secnumdepth` が `-1` より大きいとき部番号を付けます。

```

(*!book&!report)
\def\@part[#1]#2{%
  \ifnum \c@secnumdepth >\m@ne
    \refstepcounter{part}%
    \addcontentsline{toc}{part}{%
      \prepartname\thepart\postpartname\hspace{1zw}#1}%
  \else
    \addcontentsline{toc}{part}{#1}%
  \fi
  \markboth{}{}%
  {\parindent\z@

```

```

\raggedright
\interlinepenalty \@M
\normalfont
\ifnum \c@secnumdepth >\m@ne
  \Large\headfont\prepartname\thepart\postpartname
  \par\nobreak
\fi
\huge \headfont #2%
\markboth{}{}\par}%
\nobreak
\vskip 3ex
\@afterheading}
<*/book&!report>

```

book および report クラスでは secnumdepth が -2 より大きいとき部番号を付けます。

```

(*book | report)
\def\@part[#1]#2{%
  \ifnum \c@secnumdepth >-2\relax
    \refstepcounter{part}%
    \addcontentsline{toc}{part}{%
      \prepartname\thepart\postpartname\hspace{1zw}#1}%
  \else
    \addcontentsline{toc}{part}{#1}%
  \fi
  \markboth{}{}\par}%
{\centering
  \interlinepenalty \@M
  \normalfont
  \ifnum \c@secnumdepth >-2\relax
    \huge\headfont \prepartname\thepart\postpartname
    \par\vskip20\jsc@empt
  \fi
  \Huge \headfont #2\par}%
\@endpart}
</book | report>

```

\@spart 番号を付けない部です。

```

<*/book&!report>
\def\@spart#1{%
  \parindent \z@ \raggedright
  \interlinepenalty \@M
  \normalfont
  \huge \headfont #1\par}%
\nobreak
\vskip 3ex
\@afterheading}
<*/book&!report>
(*book | report)
\def\@spart#1{%

```

```

\centering
\interlinepenalty \@M
\normalfont
\Huge \headfont #1\par}%
\@endpart}
</book | report>

```

`\@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。

[2016-12-13] `openany` のときには白ページが追加されるのは変なので、その場合は追加しないようにしました。このバグは L^AT_EX では `classes.dtx` v1.4b (2000/05/19) で修正されています。

```

< *book | report>
\def\@endpart{\vfil\newpage
\if@twoside
\if@openleft %% added (2017/02/24)
\null\thispagestyle{empty}\newpage
\else\if@openright %% added (2016/12/13)
\null\thispagestyle{empty}\newpage
\fi\fi %% added (2016/12/13, 2017/02/24)
\fi
\if@restonecol
\twocolumn
\fi}
</book | report>

```

■ 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。

```

< *book | report>
\newcommand{\chapter}{%
\if@openleft\cleardoublepage\else
\if@openright\cleardoublepage\else\clearpage\fi\fi
\plainifnotempty % 元: \thispagestyle{plain}
\global\@topnum\z@
\if@english \@afterindentfalse \else \@afterindenttrue \fi
\secdef
{\@omit@numberfalse\@chapter}%
{\@omit@numbertrue\@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。

```

\def\@chapter[#1]#2{%
\ifnum \c@secnumdepth > \m@ne
<book> \if@mainmatter
\refstepcounter{chapter}%

```

```

\typeout{\@chapapp\thechapter\@chappos}%
\addcontentsline{toc}{chapter}%
{\protect\numberline
% {\if@english\thechapter\else\@chapapp\thechapter\@chappos\fi}%
{\@chapapp\thechapter\@chappos}%
#1}%
<book> \else\addcontentsline{toc}{chapter}{#1}\fi
\else
\addcontentsline{toc}{chapter}{#1}%
\fi
\chaptermark{#1}%
\addtocontents{lof}{\protect\addvspace{10\jsc@mpt}}%
\addtocontents{lot}{\protect\addvspace{10\jsc@mpt}}%
\if@twocolumn
\@topnewpage[\@makechapterhead{#2}]%
\else
\@makechapterhead{#2}%
\@afterheading
\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。

```

\def\@makechapterhead#1{%
\vspace*{2\Cvs}% 欧文は 50pt
{\parindent \z@ \raggedright \normalfont
\ifnum \c@secnumdepth >\m@ne
<book> \if@mainmatter
\huge\headfont \@chapapp\thechapter\@chappos
\par\nobreak
\vskip \Cvs % 欧文は 20pt
<book> \fi
\fi
\interlinepenalty\@M
\Huge \headfont #1\par\nobreak
\vskip 3\Cvs}} % 欧文は 40pt

```

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

```

\def\@schapter#1{%
\chaptermark{#1}%
\if@twocolumn
\@topnewpage[\@makeschapterhead{#1}]%
\else
\@makeschapterhead{#1}\@afterheading
\fi}

```

\@makeschapterhead 番号なしの章見出しです。

```

\def\@makeschapterhead#1{%
\vspace*{2\Cvs}% 欧文は 50pt
{\parindent \z@ \raggedright
\normalfont

```

```

\interlinepenalty\@M
\Huge \headfont #1\par\nobreak
\vskip 3\Cvs}} % 欧文は 40pt
</book|report>

```

■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。

```

\if@twocolumn
  \newcommand{\section}{%
    \jspf\ifx\maketitle\relax\else\maketitle\fi
    \@startsection{section}{1}{\z@}%
    (!kiyou) {0.6\Cvs}{0.4\Cvs}%
    <kiyou) {\Cvs}{0.5\Cvs}%
    % {\normalfont\large\headfont\@secapp}%
    {\normalfont\large\headfont\raggedright}}
\else
  \newcommand{\section}{%
    \if@slide\clearpage\fi
    \@startsection{section}{1}{\z@}%
    {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
    {.5\Cvs \@plus.3\Cdp}% 後アキ
    % {\normalfont\Large\headfont\@secapp}%
    {\normalfont\Large\headfont\raggedright}}
\fi

```

`\subsection` 同上です。

```

\if@twocolumn
  \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
    {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
    {\normalfont\normalsize\headfont}}
\else
  \newcommand{\subsection}{\@startsection{subsection}{2}{\z@}%
    {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
    {.5\Cvs \@plus.3\Cdp}% 後アキ
    {\normalfont\large\headfont}}
\fi

```

`\subsubsection` [2016-07-22] slide オプション指定時に `\subsubsection` の文字列と罫線が重なる問題に対処しました (forum:1982)。

```

\if@twocolumn
  \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%
    {\z@}{\if@slide .4\Cvs \else \z@ \fi}%
    {\normalfont\normalsize\headfont}}
\else
  \newcommand{\subsubsection}{\@startsection{subsubsection}{3}{\z@}%

```



```

{\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
{\if@slide .5\Cvs \@plus.3\Cdp \else \z@ \fi}%
{\normalfont\normalsize\headfont}}
\fi

```

`\paragraph` 見出しの後ろで改行されません。

[2016-11-16] 従来は `\paragraph` の最初に出るマークを「■」に固定していましたが、このマークを変更可能にするため `\jsParagraphMark` というマクロに切り出しました。これで、たとえば

```
\renewcommand{\jsParagraphMark}{★}
```

とすれば「★」に変更できますし、マークを空にすることも容易です。なお、某学会クラスでは従来どおりマークは付きません。

```

(!jspf)\newcommand{\jsParagraphMark}{■}
\if@twocolumn
  \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
    {\z@}{\if@slide .4\Cvs \else -1zw\fi}% 改行せず 1zw のアキ
    \jpf { \normalfont\normalsize\headfont}}
    (!jspf) { \normalfont\normalsize\headfont\jsParagraphMark}}
  \else
    \newcommand{\paragraph}{\@startsection{paragraph}{4}{\z@}%
      {0.5\Cvs \@plus.5\Cdp \@minus.2\Cdp}%
      {\if@slide .5\Cvs \@plus.3\Cdp \else -1zw\fi}% 改行せず 1zw のアキ
      \jpf { \normalfont\normalsize\headfont}}
      (!jspf) { \normalfont\normalsize\headfont\jsParagraphMark}}
  \fi

```

`\subparagraph` 見出しの後ろで改行されません。

```

\if@twocolumn
  \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
    {\z@}{\if@slide .4\Cvs \@plus.3\Cdp \else -1zw\fi}%
    {\normalfont\normalsize\headfont}}
  \else
    \newcommand{\subparagraph}{\@startsection{subparagraph}{5}{\z@}%
      {\z@}{\if@slide .5\Cvs \@plus.3\Cdp \else -1zw\fi}%
      {\normalfont\normalsize\headfont}}
  \fi

```

8.3 リスト環境

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 `\leftmargin k` に設定します。

`\leftmargini` 二段組であるかないかに応じてそれぞれ 2em, 2.5em でしたが、ここでは全角幅の 2 倍にしました。

[2002-05-11] 3zw に変更しました。

[2005-03-19] 二段組は 2zw に戻しました。

```
\if@slide
  \setlength\leftmargini{1zw}
\else
  \if@twocolumn
    \setlength\leftmargini{2zw}
  \else
    \setlength\leftmargini{3zw}
  \fi
\fi
```

`\leftmarginii` ii, `\leftmarginiii` iii, `\leftmarginiv` iv は `\labelsep` とそれぞれ ‘(m)’, ‘vii.’, ‘M.’ の幅との和より大きくすることになっています。ここでは全角幅の整数倍に丸めました。

```
\leftmarginiv \if@slide
\leftmarginv  \setlength\leftmarginii {1zw}
\leftmarginvi \setlength\leftmarginiii{1zw}
               \setlength\leftmarginiv {1zw}
               \setlength\leftmarginv  {1zw}
               \setlength\leftmarginvi {1zw}
\else
  \setlength\leftmarginii {2zw}
  \setlength\leftmarginiii{2zw}
  \setlength\leftmarginiv {2zw}
  \setlength\leftmarginv  {1zw}
  \setlength\leftmarginvi {1zw}
\fi
```

`\labelsep` `\labelsep` はラベルと本文の間の距離です。`\labelwidth` はラベルの幅です。これは二分
`\labelwidth` に変えました。

```
\setlength \labelsep {0.5zw} % .5em
\setlength \labelwidth{\leftmargini}
\addtolength\labelwidth{-\labelsep}
```

`\partopsep` リスト環境の前に空行がある場合、`\parskip` と `\topsep` に `\partopsep` を加えた値だけ縦方向の空白ができます。0 に改変しました。

```
\setlength\partopsep{\z@} % {2\p@ \@plus 1\p@ \@minus 1\p@}
```

`\@beginparpenalty` リストや段落環境の前後、リスト項目間に挿入されるペナルティです。

```
\@endparpenalty  \@beginparpenalty -\@lowpenalty
\@itempenalty    \@endparpenalty    -\@lowpenalty
                 \@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` の中では小さい値に設定されます）。このため、`\normalsize` がすべてのパラメータを戻せるように、`\@listI` で `\@listi` のコピーを保存します。元の値はかなり複雑ですが、ここでは簡素化してしまいました。特に最初と最後に行送りの半分の空きが入るようにしてあります。

す。アスキーの標準スタイルではトップレベルの `itemize`, `enumerate` 環境でだけ最初と最後に行送りの半分の空きが入るようになっていました。

[2004-09-27] `\topsep` のグルー $\pm_{0.1}^{0.2}$ `\baselineskip` を思い切って外しました。

```
\def\@listi{\leftmargin\leftmarginii
\parsep \z@
\topsep 0.5\baselineskip
\itemsep \z@ \relax}
\let\@listI\@listi
```

念のためパラメータを初期化します（実際には不要のようです）。

```
\@listi
```

`\@listii` 第 2～6 レベルのリスト環境のパラメータの設定です。

```
\@listiii \def\@listii{\leftmargin\leftmarginii
\labelwidth\leftmarginii \advance\labelwidth-\labelsep
\@listiv \topsep \z@
\@listv \parsep \z@
\@listvi \itemsep\parsep}
\def\@listiii{\leftmargin\leftmarginiii
\labelwidth\leftmarginiii \advance\labelwidth-\labelsep
\topsep \z@
\parsep \z@
\itemsep\parsep}
\def\@listiv {\leftmargin\leftmarginiv
\labelwidth\leftmarginiv
\advance\labelwidth-\labelsep}
\def\@listv {\leftmargin\leftmarginv
\labelwidth\leftmarginv
\advance\labelwidth-\labelsep}
\def\@listvi {\leftmargin\leftmarginvi
\labelwidth\leftmarginvi
\advance\labelwidth-\labelsep}
```

■**enumerate 環境** `enumerate` 環境はカウンタ `enumi`, `enumii`, `enumiii`, `enumiv` を使います。`enumn` 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力する命令です。

```
\renewcommand{\theenumi}{\@arabic\c@enumi}
\renewcommand{\theenumii}{\@alph\c@enumii}
\renewcommand{\theenumiii}{\@roman\c@enumiii}
\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
`\labelenumii`
`\labelenumiii`
`\labelenumiv`

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。

```
\newcommand{\labelenumi}{\theenumi.}
\newcommand{\labelenumii}{\inhibitglue (\theenumii) \inhibitglue}
\newcommand{\labelenumiii}{\theenumiii.}
\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
`\p@enumiii` 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 \renewcommand{\p@enumii}{\theenumi}
\renewcommand{\p@enumiii}{\theenumi\inhibitglue (\theenumii) }
\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 \newcommand{\labelitemi}{\textbullet}
\labelitemiii \newcommand{\labelitemii}{\normalfont\bfseries \textendash}
\labelitemiv \newcommand{\labelitemiii}{\textasteriskcentered}
\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。

```
\newenvironment{description}{%
\list{}{%
\labelwidth=\leftmargin
\labelsep=1zw
\advance \labelwidth by -\labelsep
\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{1zw}`) を入れるのもいいと思います。

```
\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。book クラスでは各章の初めにちょっとしたことを書くのに使います。titlepage オプション付きの article クラスでは、独立したページに出力されます。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。

JSPF スタイルでは実際の出力は `\maketitle` で行われます。

```
(<*book>
\newenvironment{abstract}{%
\begin{list}{}{%
\listparindent=1zw
```

```

\itemindent=\listparindent
\rightmargin=0pt
\leftmargin=5zw\item[]{}\end{list}\vspace{\baselineskip}}
{/book}
<*article | report | kiyou>
\newbox\@abstractbox
\if@titlepage
\newenvironment{abstract}{%
\titlename
\null\vfil
\@beginparpenalty\@lowpenalty
\begin{center}%
\headfont \abstractname
\@endparpenalty\@M
\end{center}}%
{\par\vfil\null\endtitlepage}
\else
\newenvironment{abstract}{%
\if@twocolumn
\ifx\maketitle\relax
\section*{\abstractname}%
\else
\global\setbox\@abstractbox\hbox\bgroup
\begin{minipage}[b]{\textwidth}
\small\parindent1zw
\begin{center}%
{\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
\end{center}%
\list{}{%
\listparindent\parindent
\itemindent \listparindent
\rightmargin \leftmargin}%
\item\relax
\fi
\else
\small
\begin{center}%
{\headfont \abstractname\vspace{-.5em}\vspace{\z@}}%
\end{center}%
\list{}{%
\listparindent\parindent
\itemindent \listparindent
\rightmargin \leftmargin}%
\item\relax
\fi}{\if@twocolumn
\ifx\maketitle\relax
\else
\endlist\end{minipage}\egroup
\fi

```

```

\else
\endlist
\fi}
\fi
</article | report | kiyou>
<*jspf>
\newbox\@abstractbox
\newenvironment{abstract}{%
\global\setbox\@abstractbox\hbox\bgroup
\begin{minipage}[b]{157\jsc@mmm}{\sffamily Abstract}\par
\small
\if@english \parindent6\jsc@mmm \else \parindent1zw \fi}%
{\end{minipage}\egroup}
</jspf>

```

■キーワード

keywords キーワードを準備する環境です。実際の出力は `\maketitle` で行われます。

```

<*jspf>
%\newbox\@keywordsbox
%\newenvironment{keywords}{%
% \global\setbox\@keywordsbox\hbox\bgroup
% \begin{minipage}[b]{1570\jsc@mmm}{\sffamily Keywords:}\par
% \small\parindent0zw}%
% {\end{minipage}\egroup}
</jspf>

```

■verse 環境

verse 詩のための `verse` 環境です。

```

\newenvironment{verse}{%
\let \=\@centercr
\list{}{%
\itemsep \z@
\itemindent -2zw % 元: -1.5em
\listparindent\itemindent
\rightmargin \z@
\advance\leftmargin 2zw}% 元: 1.5em
\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。

```

\newenvironment{quotation}{%
\list{}{%
\listparindent\parindent
\itemindent\listparindent

```

```

\rightmargin \z0}%
\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。

```

\newenvironment{quote}%
{\list{}{\rightmargin\z0}\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 1zw にし、括弧を全角にしました。

```

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

```

titlepage タイトルを独立のページに出力するのに使われます。

[2017-02-24] コミュニティ版 pL^AT_EX の標準クラス 2017/02/15 に合わせて、book クラスでタイトルを必ず奇数ページに送るようにしました。といっても、横組クラスしかありませんでしたので、従来の挙動は何も変わっていません。また、book 以外の場合のページ番号のリセットもコミュニティ版 pL^AT_EX の標準クラス 2017/02/15 に合わせましたが、こちらでも片面印刷あるいは独立のタイトルページを作らないクラスばかりでしたので、従来の挙動は何も変わらずに済みました。

```

\newenvironment{titlepage}{%
(book) \pltx@cleartooddpage %% 2017-02-24
\if@twocolumn
\@restonecoltrue\onecolumn
\else
\@restonecolfalse\newpage
\fi
\thispagestyle{empty}%
\ifodd\c@page\setcounter{page}\@ne\else\setcounter{page}\z@\fi %% 2017-02-24
}%
{\if@restonecol\twocolumn \else \newpage \fi
\if@twoside\else
\setcounter{page}\@ne
\fi}

```

■付録

`\appendix` 本文と付録を分離するコマンドです。

```
(*!book&!report)
\newcommand{\appendix}{\par
  \setcounter{section}{0}%
  \setcounter{subsection}{0}%
  \gdef\presectionname{\appendixname}%
  \gdef\postsectionname{}%
  % \gdef\thesection{\@Alph\c@section}% [2003-03-02]
  \gdef\thesection{\presectionname\@Alph\c@section\postsectionname}%
  \gdef\thesubsection{\@Alph\c@section.\@arabic\c@subsection}}
(!book&!report)
(*book | report)
\newcommand{\appendix}{\par
  \setcounter{chapter}{0}%
  \setcounter{section}{0}%
  \gdef\@chapapp{\appendixname}%
  \gdef\@chappos{}%
  \gdef\thechapter{\@Alph\c@chapter}}
(/book | report)
```

8.4 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 倍の幅の空きが入ります。

```
\setlength\arraycolsep{5\jsc@empt}
```

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

```
\setlength\tabcolsep{6\jsc@empt}
```

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

```
\setlength\arrayrulewidth{.4\jsc@empt}
```

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

```
\setlength\doublerulesep{2\jsc@empt}
```

■tabbing 環境

`\tabbingsep` \’ コマンドで入るアキです。

```
\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。

```
\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きです。

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

```
\setlength\fboxsep{3\jsc@mp}{  
\setlength\fboxrule{.4\jsc@mp}
```

■equation と eqnarray 環境

`\theequation` 数式番号を出力するコマンドです。

```
(!book&!report)\renewcommand \theequation {\@arabic\c@equation}  
(*book|report)  
\@addtoreset{equation}{chapter}  
\renewcommand\theequation  
{\ifnum \c@chapter>z@ \thechapter.\fi \@arabic\c@equation}  
(/book|report)
```

`\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。

```
% \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。

```
% \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。

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

8.5 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。

```
<!*book&!report>
\newcounter{figure}
\renewcommand \thefigure {\@arabic\c@figure}
</!book&!report>
<*book | report>
\newcounter{figure}[chapter]
\renewcommand \thefigure
  {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
</book | report>
```

`\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 \def\fps@figure{tbp}
\ftype@figure \def\ftype@figure{1}
\fnum@figure \def\ext@figure{lof}
\def\fnum@figure{\figurename\nobreak\thefigure}
```

`figure` * 形式は段抜きのフロートです。

```
figure* \newenvironment{figure}%
        {\@float{figure}}%
        {\end@float}
\newenvironment{figure*}%
        {\@dblfloat{figure}}%
        {\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。

```
<!*book&!report>
\newcounter{table}
\renewcommand\thetable{\@arabic\c@table}
</!book&!report>
<*book | report>
\newcounter{table}[chapter]
\renewcommand \thetable
  {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
</book | report>
```

`\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 \def\fps@table{tbp}
\ftype@table \def\ftype@table{2}
\fnum@table \def\ext@table{lot}
\def\fnum@table{\tablename\nobreak\thetable}
```

`table` * は段抜きのフロートです。

`table*`

```

\newenvironment{table}%
    {\@float{table}}%
    {\end@float}
\newenvironment{table*}%
    {\@dblfloat{table}}%
    {\end@dblfloat}

```

8.6 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。

```

\newlength\abovecaptionskip
\newlength\belowcaptionskip
\setlength\abovecaptionskip{5\jsc@mp} % 元: 10\p@
\setlength\belowcaptionskip{5\jsc@mp} % 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。

```

<*/jspf>
% \long\def\@makecaption#1#2{{\small
%   \advance\leftskip 10\jsc@mmm
%   \advance\rightskip 10\jsc@mmm
%   \vskip\abovecaptionskip
%   \sbox\@tempboxa{#1\hskip1zw\relax #2}%
%   \ifdim \wd\@tempboxa >\hsize
%     #1\hskip1zw\relax #2\par
%   \else
%     \global \minipagefalse
%     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
%   \fi
%   \vskip\belowcaptionskip}}
\long\def\@makecaption#1#2{{\small
  \advance\leftskip .0628\linewidth
  \advance\rightskip .0628\linewidth
  \vskip\abovecaptionskip
  \sbox\@tempboxa{#1\hskip1zw\relax #2}%
  \ifdim \wd\@tempboxa <\hsize \centering \fi
  #1\hskip1zw\relax #2\par
  \vskip\belowcaptionskip}}
</jspf>
<*/jspf>
\long\def\@makecaption#1#2{%

```

```

\vskip\abovecaptionskip
\sbbox\@tempboxa{\small\sffamily #1\quad #2}%
\ifdim \wd\@tempboxa >\hsize
  {\small\sffamily
   \list{#1}{%
    \renewcommand{\makelabel}[1]{##1\hfil}
    \itemsep \z@
    \itemindent \z@
    \labelsep \z@
    \labelwidth 11\jsc@mmm
    \listparindent\z@
    \leftmargin 11\jsc@mmm}\item\relax #2\endlist}
  \else
    \global \@minipagefalse
    \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
  \fi
\vskip\belowcaptionskip}
</jspf>

```

9 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 \DeclareOldFontCommand{\mc}{\normalfont\mcfamily}{\mathmc}
\rm \DeclareOldFontCommand{\gt}{\normalfont\gtfamily}{\mathgt}
\sf \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\tt \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\tt \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すコマンドは `\mdseries` です。

```
\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すコマンドは `\upshape` です。

```

\it \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\sl \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\sc \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 \DeclareRobustCommand*{\cal}{\@fontswitch\relax\mathcal}
\mit \DeclareRobustCommand*{\mit}{\@fontswitch\relax\mathnormal}

```

10 相互参照

10.1 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` \geq `\@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。

```
\newcommand\@pnumwidth{1.55em}
\newcommand\@tocrmarg{2.55em}
\newcommand\@dotsep{4.5}
<!book&!report>\setcounter{tocdepth}{2}
(book | report)\setcounter{tocdepth}{1}
```

■目次

`\tableofcontents` 目次を生成します。

`\jsc@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)

```
\newdimen\jsc@tocl@width
\newcommand{\tableofcontents}{%
(*book | report)
  \settowidth\jsc@tocl@width{\headfont\prechaptername\postchaptername}%
  \settowidth\@tempdima{\headfont\appendixname}%
  \ifdim\jsc@tocl@width<\@tempdima \setlength\jsc@tocl@width{\@tempdima}\fi
  \ifdim\jsc@tocl@width<2zw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1zw\fi
  \if@twocolumn
    \@restonecoltrue\onecolumn
  \else
    \@restonecolfalse
  \fi
  \chapter*{\contentsname}%
  \@mkboth{\contentsname}{}%
(/book | report)
(*!book&!report)
  \settowidth\jsc@tocl@width{\headfont\presectionname\postsectionname}%
  \settowidth\@tempdima{\headfont\appendixname}%
  \ifdim\jsc@tocl@width<\@tempdima\relax\setlength\jsc@tocl@width{\@tempdima}\fi
  \ifdim\jsc@tocl@width<2zw \divide\jsc@tocl@width by 2 \advance\jsc@tocl@width 1zw\fi
  \section*{\contentsname}%
  \@mkboth{\contentsname}{\contentsname}%
(/!book&!report)
  \@starttoc{toc}%
(book | report) \if@restonecol\twocolumn\fi
}
```

`\l@part` 部の目次です。

```
\newcommand*{\l@part}[2]{%
  \ifnum \c@tocdepth >-2\relax
  (!book&!report) \addpenalty\@secpenalty
  (book | report) \addpenalty{-\@highpenalty}%
    \addvspace{2.25em \@plus\jsc@empt}%
    \begingroup
      \parindent \z@
%    \@pnumwidth should be \@tocrmarg
%    \rightskip \@pnumwidth
    \rightskip \@tocrmarg
    \parfillskip -\rightskip
    {\leavevmode
      \large \headfont
      \setlength\@lnumwidth{4zw}%
      #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
    \nobreak
  (book | report) \global\@nobreaktrue
  (book | report) \everypar{\global\@nobreakfalse\everypar{}}%
```

```

\endgroup
\fi}

```

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

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

```

(*book|report)
\newcommand*{\l@chapter}[2]{%
  \ifnum \c@tocdepth >\m@ne
    \addpenalty{-\@highpenalty}%
    \addvspace{1.0em \@plus\jsc@empt}%
%   \vskip 1.0em \@plus\p@ % book.cls では↑がこうなっている
    \begingroup
      \parindent\z@
%     \rightskip\@pnumwidth
      \rightskip\@tocrmarg
      \parfillskip-\rightskip
      \leavevmode\headfont
      % \if@english\setlength\@l@numwidth{5.5em}\else\setlength\@l@numwidth{4.683zw}\fi
      \setlength\@l@numwidth{\jsc@tocl@width}\advance\@l@numwidth 2.683zw
      \advance\leftskip\@l@numwidth \hskip-\leftskip
      #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
      \penalty\@highpenalty
    \endgroup
  \fi}
{/book|report)

```

\l@section 節の目次です。

```

(*!book&!report)
\newcommand*{\l@section}[2]{%
  \ifnum \c@tocdepth >\z@
    \addpenalty{\@secpenalty}%
    \addvspace{1.0em \@plus\jsc@empt}%
    \begingroup
      \parindent\z@
%     \rightskip\@pnumwidth
      \rightskip\@tocrmarg
      \parfillskip-\rightskip
      \leavevmode\headfont
      %\setlength\@l@numwidth{4zw}% 元 1.5em [2003-03-02]
      \setlength\@l@numwidth{\jsc@tocl@width}\advance\@l@numwidth 2zw
      \advance\leftskip\@l@numwidth \hskip-\leftskip
      #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
    \endgroup
  \fi}
{/!book&!report)

```

インデントと幅はそれぞれ 1.5em, 2.3em でしたが, 1zw, 3.683zw に変えました。

(book|report) % \newcommand*{\l@section}{\@dottedtocline{1}{1zw}{3.683zw}}

[2013-12-30] 上のインデントは \jsc@tocl@width から決めるようにしました。(by ts)

\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、要修正かもしれせん。

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

```
\l@subsubsection
\l@paragraph
\l@subparagraph

(*!book&!report)
% \newcommand*\l@section{\@dottedtocline{2}{1.5em}{2.3em}}
% \newcommand*\l@subsection{\@dottedtocline{3}{3.8em}{3.2em}}
% \newcommand*\l@subsubsection{\@dottedtocline{4}{7.0em}{4.1em}}
% \newcommand*\l@paragraph{\@dottedtocline{5}{10em}{5em}}
%
% \newcommand*\l@subparagraph{\@dottedtocline{2}{1zw}{3zw}}
% \newcommand*\l@subsubsection{\@dottedtocline{3}{2zw}{3zw}}
% \newcommand*\l@paragraph{\@dottedtocline{4}{3zw}{3zw}}
% \newcommand*\l@subparagraph{\@dottedtocline{5}{4zw}{3zw}}
%
\newcommand*\l@section{%
    \@tempdima\jsc@tocl@width \advance\@tempdima -1zw
    \@dottedtocline{2}{\@tempdima}{3zw}}
\newcommand*\l@subsection{%
    \@tempdima\jsc@tocl@width \advance\@tempdima 0zw
    \@dottedtocline{3}{\@tempdima}{4zw}}
\newcommand*\l@subsubsection{%
    \@tempdima\jsc@tocl@width \advance\@tempdima 1zw
    \@dottedtocline{4}{\@tempdima}{5zw}}
\newcommand*\l@paragraph{%
    \@tempdima\jsc@tocl@width \advance\@tempdima 2zw
    \@dottedtocline{5}{\@tempdima}{6zw}}
(!book&!report)
(*book | report)
% \newcommand*\l@section{\@dottedtocline{2}{3.8em}{3.2em}}
% \newcommand*\l@subsection{\@dottedtocline{3}{7.0em}{4.1em}}
% \newcommand*\l@subsubsection{\@dottedtocline{4}{10em}{5em}}
% \newcommand*\l@paragraph{\@dottedtocline{5}{12em}{6em}}
\newcommand*\l@section{%
    \@tempdima\jsc@tocl@width \advance\@tempdima -1zw
    \@dottedtocline{1}{\@tempdima}{3.683zw}}
\newcommand*\l@subsection{%
    \@tempdima\jsc@tocl@width \advance\@tempdima 2.683zw
    \@dottedtocline{2}{\@tempdima}{3.5zw}}
\newcommand*\l@subsubsection{%
    \@tempdima\jsc@tocl@width \advance\@tempdima 6.183zw
    \@dottedtocline{3}{\@tempdima}{4.5zw}}
\newcommand*\l@paragraph{%
    \@tempdima\jsc@tocl@width \advance\@tempdima 10.683zw
    \@dottedtocline{4}{\@tempdima}{5.5zw}}
\newcommand*\l@subparagraph{%
    \@tempdima\jsc@tocl@width \advance\@tempdima 16.183zw
```



```

\@dottedtocline{5}{\@tempdima}{6.5zw}}
</book | report>

```

`\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。

```

\newdimen\@lnumwidth
\def\numberline#1{\hb@xt@\@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。

```

\def\@dottedtocline#1#2#3#4#5{\ifnum #1>\c@tocdepth \else
\vskip \z@ \@plus.2\jcs@ompt
{\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
\parindent #2\relax\@afterindenttrue
\interlinepenalty\@M
\leavevmode
\@lnumwidth #3\relax
\advance\leftskip \@lnumwidth \null\nobreak\hskip -\leftskip
{#4}\nobreak
\leaders\hbox{$\m@th \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
mu$}\hfill \nobreak\hb@xt@\@pnumwidth{%
\hfil\normalfont \normalcolor #5}\par}\fi}

```

■ 図目次と表目次

`\listoffigures` 図目次を出力します。

```

\newcommand{\listoffigures}{%
(*book | report)
\if@twocolumn\@restonecoltrue\onecolumn
\else\@restonecolfalse\fi
\chapter*{\listfigurename}%
\@mkboth{\listfigurename}{}%
</book | report>
(*!book&!report)
\section*{\listfigurename}%
\@mkboth{\listfigurename}{\listfigurename}%
</!book&!report>
\@starttoc{lof}%
(book | report) \if@restonecol\twocolumn\fi
}

```

`\l@figure` 図目次の項目を出力します。

```

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

```

`\listoftables` 表目次を出力します。

```

\newcommand{\listoftables}{%

```

```

(*book | report)
  \if@twocolumn\@restonecoltrue\onecolumn
  \else\@restonecolfalse\fi
  \chapter*{\listtablename}%
  \mkboth{\listtablename}{}%
</book | report>
(*!book&!report)
  \section*{\listtablename}%
  \mkboth{\listtablename}{\listtablename}%
</!book&!report>
  \starttoc{lot}%
<book | report> \if@restonecol\twocolumn\fi
}

```

`\l@table` 表目次は図目次と同じです。

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

10.2 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。

```

\newdimen\bibindent
\setlength\bibindent{2zw}

```

`thebibliography` 参考文献リストを出力します。

```

\newenvironment{thebibliography}[1]{%
  \global\let\presectionname\relax
  \global\let\postsectionname\relax
<article | jspf> \section*{\refname}\mkboth{\refname}{\refname}%
<*kiyou>
  \vspace{1.5\baselineskip}
  \subsubsection*{\refname}\mkboth{\refname}{\refname}%
  \vspace{0.5\baselineskip}
</kiyou>
<book | report> \chapter*{\bibname}\mkboth{\bibname}{}%
<book | report> \addcontentsline{toc}{chapter}{\bibname}%
  \list{\@biblabel{\@arabic\c@enumiv}}%
    {\settowidth\labelwidth{\@biblabel{#1}}%
     \leftmargin\labelwidth
     \advance\leftmargin\labelsep
     \@openbib@code
     \usecounter{enumiv}%
     \let\p@enumiv\@empty
     \renewcommand\theenumiv{\@arabic\c@enumiv}}%
<kiyou> \small
  \sloppy
  \clubpenalty4000
  \@clubpenalty\clubpenalty
  \widowpenalty4000%

```

```

\sfcode'\.\@m}
{\def\@noitemerr
  {\@latex@warning{Empty 'thebibliography' environment}}}%
\endlist}

\newblock \newblock はデフォルトでは小さなスペースを生成します。
\newcommand{\newblock}{\hskip .11em\@plus.33em\@minus.07em}

\@openbib@code \@openbib@code はデフォルトでは何もしません。この定義は openbib オプションによっ
て変更されます。
\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。
% \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。かつ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。
% \def\@citex[#1]#2{\leavevmode
%   \let\@citea\@empty
%   \@cite{\@for\@citeb:=#2\do
%     {\@citea\def\@citea{, \inhibitglue\penalty\@m\ }%
%       \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}}%
%     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
%     \@ifundefined{b@\@citeb}{\mbox{\normalfont\bfseries ?}}%
%       \G@refundefinedtrue
%       \@latex@warning
%         {Citation '\@citeb' on page \thepage \space undefined}}%
%     {\@citeo{ofmt{\csname b@\@citeb\endcsname}}}\@citeb}}%
% \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。
% \DeclareRobustCommand\cite{\unskip
%   \@ifnextchar [{\@tempswattrue\@citex}{\@tempswafalse\@citex[]}]
%   \def\@cite#1#2{\$~{\hbox{\scriptsize{#1\if@tempswa
%     , \inhibitglue\ #2\fi}}}}\$}

```

10.3 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）。

```
\newenvironment{theindex}{% 索引を 3 段組で出力する環境
```

```

\if@twocolumn
  \onecolumn\@restonecolfalse
\else
  \clearpage\@restonecoltrue
\fi
\columnseprule.4pt \columnsep 2zw
\ifx\multicols\@undefined
<book | report>      \twocolumn[\@makeschapterhead{\indexname}%
<book | report>      \addcontentsline{toc}{chapter}{\indexname}}%
<!book&!report>      \def\presectionname{}\def\postsectionname{}%
<!book&!report>      \twocolumn[{\section*{\indexname}}%
  \else
    \ifdim\textwidth<\fullwidth
      \setlength{\evensidemargin}{\oddsidemargin}
      \setlength{\textwidth}{\fullwidth}
      \setlength{\linewidth}{\fullwidth}
<book | report>      \begin{multicols}{3}[{\chapter*{\indexname}}%
<book | report>      \addcontentsline{toc}{chapter}{\indexname}}%
<!book&!report>      \def\presectionname{}\def\postsectionname{}%
<!book&!report>      \begin{multicols}{3}[{\section*{\indexname}}%
    \else
<book | report>      \begin{multicols}{2}[{\chapter*{\indexname}}%
<book | report>      \addcontentsline{toc}{chapter}{\indexname}}%
<!book&!report>      \def\presectionname{}\def\postsectionname{}%
<!book&!report>      \begin{multicols}{2}[{\section*{\indexname}}%
    \fi
  \fi
<book | report>      \@mkboth{\indexname}{}%
<!book&!report>      \@mkboth{\indexname}{\indexname}%
    \plainifnotempty % \thispagestyle{plain}
    \parindent\z@
    \parskip\z@ \@plus .3\jcs@mpt\relax
    \let\item\@idxitem
    \raggedright
    \footnotesize\narrowbaselines
  }{
    \ifx\multicols\@undefined
      \if@restonecol\onecolumn\fi
    \else
      \end{multicols}
    \fi
    \clearpage
  }

```

\@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 \newcommand{\@idxitem}{\par\hangindent 4zw} % 元 40pt

\subsubitem \newcommand{\subitem}{\@idxitem \hspace*{2zw}} % 元 20pt

\subsubitem \newcommand{\subsubitem}{\@idxitem \hspace*{3zw}} % 元 30pt

`\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。

```
\newcommand{\indexspace}{\par \vskip 10\jsc@empt \@plus5\jsc@empt \@minus3\jsc@empt\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*,

`\alsoname` *see also* 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。

```
\newcommand\seename{\if@english see\else →\fi}
```

```
\newcommand\alsoname{\if@english see also\else →\fi}
```

10.4 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。pL^AT_EX の日付が 2016/09/03 より新しい場合は, このパッチが不要なのであてません。

```
\@ifl@t@r\pfmtversion{2016/09/03}
{\jsc@needspace@tchfalse}{\jsc@needspace@tchtrue}
\ifjsc@needspace@tch
\let\footnotes@ve=\footnote
\def\footnote{\inhibitglue\footnotes@ve}
\let\footnotemarks@ve=\footnotemark
\def\footnotemark{\inhibitglue\footnotemarks@ve}
\fi
```

`\@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 `\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。

[2002-04-09] インプリメントの仕方を変えたため消しました。

[2013-04-23] 新しい pT_EX では脚注番号のまわりにスペースが入りすぎることを防ぐため, 北川さんのパッチ [qa:57090] を取り込みました。

[2013-05-14] plcore.ltx に倣った形に書き直しました (Thanks: 北川さん)。

[2016-07-11] コミュニティ版 pL^AT_EX の変更に従いました (Thanks: 角藤さん)。pL^AT_EX の日付が 2016/04/17 より新しい場合は, このパッチが不要なのであてません。

```
\@ifl@t@r\pfmtversion{2016/04/17}
{\jsc@needspace@tchfalse}{\jsc@needspace@tchtrue}
\ifjsc@needspace@tch
\renewcommand\@makefnmark{%
\ifdir \hbox{\hbox{\@textsuperscript{\normalfont\@thefnmark}}\hbox{}}%
\else\hbox{\yoko\@textsuperscript{\normalfont\@thefnmark}}\fi}
\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。

[2003-08-15] `\textasteriskcentered` ではフォントによって下がりすぎるので変更しました。

[2016-10-08] TODO: 脚注番号が `newtxtext` や `newpxtext` の使用時におかしくなっています。これらのパッケージは内部で `\thefootnote` を再定義していますので、気になる場合はパッケージを読み込むときに `defaultsup` オプションを付けてください (qa:57284, qa:57287)。

```
\def\thefootnote{\ifnum\c@footnote>\z@\leavevmode\lower.5ex\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。

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

`\footnoterule` 本文と脚注の間の罫線です。

```
\renewcommand{\footnoterule}{%
  \kern-3\jsc@empt
  \hrule width .4\columnwidth height 0.4\jsc@empt
  \kern 2.6\jsc@empt}
```

`\c@footnote` 脚注番号は章ごとにリセットされます。

```
(book | report)\@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)

[2016-08-25] コミュニティ版 p_LA_TE_X の「閉じ括弧類の直後に `\footnotetext` が続く場合に改行が起きることがある問題に対処」と同等のコードを追加しました。

[2016-09-08] コミュニティ版 p_LA_TE_X のバグ修正に追随しました。

[2016-11-29] 古い p_LA_TE_X で使用された場合を考慮してコードを改良。

```
\long\def\@footnotetext{%
  \insert\footins\bgroup
  \normalfont\footnotesize
  \interlinepenalty\interfootnotelinepenalty
  \splittopskip\footnotesep
  \splitmaxdepth \dp\strutbox \floatingpenalty \MM
  \hsize\columnwidth \@parboxrestore
  \protected@edef\@currentlabel{%
    \csname p@footnote\endcsname\@thefnmark
  }%
  \color@begingroup
  \@makefnmark{%
    \rule{\footnotesep}{\footnotesep}\ignorespaces}%
  \futurelet\next\fo@t}
\def\fo@t{\ifcat\bgroup\noexpand\next \let\next\fo@@t
  \else \let\next\fo@t\fi \next}
\def\fo@@t{\bgroup\aftergroup\@foot\let\next}
\def\fo@t#1{#1\@foot}
\def\@foot{\@finalstrut\strutbox\color@endgroup\egroup
  \ifx\pltx@foot@penalty\undefined\else
    \ifhmode\null\fi
    \ifnum\pltx@foot@penalty=\z@\else
      \penalty\pltx@foot@penalty
```

```

\pltx@foot@penalty\z@
\fi
\fi}

```

`\@makefntext` 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。

```

\newcommand\@makefntext[1]{%
  \advance\leftskip 3zw
  \parindent 1zw
  \noindent
  \llap{\@makefnmark\hskip0.3zw}#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。

[2002-04-09] インプリメントの仕方を変えたため消しました。

```

% \def\@xfootnotenext[#1]{%
%   \begingroup
%     \ifnum#1>\z@
%       \csname c@\@mpfn\endcsname #1\relax
%       \unrestored@protected@xdef\@thefnmark{\thempfn}%
%     \else
%       \unrestored@protected@xdef\@thefnmark{}%
%     \fi
%   \endgroup
%   \@footnotetext}

```

11 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。

[2016-07-18] `\inhibitglue` の発行対象を `\inhibitxspcode` が 2 に設定されているものすべてに拡大しました。

[2016-12-01] すぐ上の変更で `\@tempa` を使っていたのがよくなかったので、プレフィックスを付けて `\jsc@tempa` にしました (forum:2085)。

[2017-02-13] `\jsc@tempa` は実はテンポラリーではなく「この処理専用のユニーク制御綴」である必要があります。間違って別の箇所で使う危険性が高いので、専用の命令 `\jsc@ig@temp` に置き換えました (Issue #54)。

```

\def\@inhibitglue{%
  \futurelet\@let@token\@@inhibitglue}
\begingroup
\let\GDEF=\gdef
\let\CATCODE=\catcode
\let\ENDGROUP=\endgroup

```

```

\CATCODE'k=12
\CATCODE'a=12
\CATCODE'n=12
\CATCODE'j=12
\CATCODE'i=12
\CATCODE'c=12
\CATCODE'h=12
\CATCODE'r=12
\CATCODE't=12
\CATCODE'e=12
\GDEF\KANJI@CHARACTER{kanji character }
\ENDGROUP
\def\@@inhibitglue{%
  \expandafter\expandafter\expandafter\jsc@inhibitglue\expandafter\meaning\expandafter\@let@to
\expandafter\def\expandafter\jsc@inhibitglue\expandafter#\expandafter1\KANJI@CHARACTER#2#3\jsc
  \def\jsc@ig@temp{#1}%
  \ifx\jsc@ig@temp@empty
    \ifnum\the\inhibitxspcode'#2=2\relax
      \inhibitglue
    \fi
  \fi}
\let\everyparhook=\@inhibitglue
\AtBeginDocument{\everypar{\everyparhook}}

```

これだけではいけないようです。あちこちに `\everypar` を初期化するコマンドが隠されていました。

まず、環境の直後の段落です。

[2016-11-19] `ltlists.dtx` 2015/05/10 v1.0t の変更に従って `\clubpenalty` のリセットを追加しました。

```

\def\@doendpe{%
  \@endpetrue
  \def\par{%
    \@restorepar\clubpenalty\@clubpenalty\everypar{\everyparhook}\par\@endpefalse}%
    \everypar{\setbox\z@\lastbox}\everypar{\everyparhook}\@endpefalse\everyparhook}}

```

[2017-08-31] `minipage` 環境にも対策します。

```

\def\@setminipage{%
  \@minipagetrue
  \everypar{\@minipagefalse\everypar{\everyparhook}}%
}

```

`\item` 命令の直後です。

```

\def\@item[#1]{%
  \if@noperitem
    \@donoperitem
  \else
    \if@inlabel
      \indent \par
    \fi

```



```

\ifhmode
  \unskip\unskip \par
\fi
\if@newlist
  \if@nobreak
    \@nbitem
  \else
    \addpenalty\@beginparpenalty
    \addvspace\@topsep
    \addvspace{-\parskip}%
  \fi
\else
  \addpenalty\@itempenalty
  \addvspace\itemsep
\fi
\global\@inlabeltrue
\fi
\everypar{%
  \@minipagefalse
  \global\@newlistfalse
  \if@inlabel
    \global\@inlabelfalse
    {\setbox\z@\lastbox
     \ifvoid\z@
       \kern-\itemindent
     \fi}%
    \box\@labels
    \penalty\z@
  \fi
  \if@nobreak
    \@nobreakfalse
    \clubpenalty \@M
  \else
    \clubpenalty \@clubpenalty
    \everypar{\everyparhook}%
  \fi\everyparhook}%
\if@noitemarg
  \@noitemargfalse
  \if@nmbrlist
    \refstepcounter\@listctr
  \fi
\fi
\sbox\@tempboxa{\makelabel{#1}}%
\global\setbox\@labels\hbox{%
  \unhbox\@labels
  \hskip \itemindent
  \hskip -\labelwidth
  \hskip -\labelsep
  \ifdim \wd\@tempboxa >\labelwidth

```

```

\box\@tempboxa
\else
\hbox to\labelwidth {\unhbox\@tempboxa}%
\fi
\hskip \labelsep}%
\ignorespaces}

```

二つ挿入した `\everyparhook` のうち後者が `\section` 類の直後に 2 回、前者が 3 回目以降に実行されます。

```

\def\@afterheading{%
\@nbreaktrue
\everypar{%
\if@nbreak
\@nbreakfalse
\clubpenalty \@M
\if@afterindent \else
{\setbox\z@\lastbox}%
\fi
\else
\clubpenalty \@clubpenalty
\everypar{\everyparhook}%
\fi\everyparhook}}

```

`\@gnewline` についてはちょっと複雑な心境です。もともとの `pLATEX 2ε` 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。

```

\def\@gnewline #1{%
\ifvmode
\@nolnerr
\else
\unskip \reserved@e {\reserved@f#1}\nbreak \hfil \break \null
\inhibitglue \ignorespaces
\fi}

```

12 いろいろなロゴ

L^AT_EX 関連のロゴを作り直します。

[2016-07-14] ロゴの定義は `jslogo` パッケージに移転しました。後方互換のため、`jsclasses` ではデフォルトでこれを読み込みます。`nojslogo` オプションが指定されている場合は読み込みません。

\小 文字を小さめに出したり上寄りに小さめに出したりする命令を、`jslogo.sty` では名称変更
 \上小 してありますので、コピーします。

```

\if@jslogo
\IfFileExists{jslogo.sty}{%

```

```

\RequirePackage{jslogo}%
\def\小{\jslg@small}%
\def\上小{\jslg@uppersmall}%
}{%
\ClassWarningNoLine{\jsc@clsname}{%
  The redefinitions of LaTeX-related logos has\MessageBreak
  been moved to jslogo.sty since 2016, but\MessageBreak
  jslogo.sty not found. Current release of\MessageBreak
  'jsclasses' includes it, so please check\MessageBreak
  the installation}%
}
\fi

```

13 amsmath との衝突の回避

`\ltx@ifnextchar` amsmath パッケージでは行列中で `\@ifnextchar` を再定義していますが、これが L^AT_EX の `\ProvidesFile` で悪さをする例が F_TE_X で報告されています。これを避けるための tDB さんのフィックスを挿入しておきます。副作用がありましたらお知らせください。

この現象については私の TeX 掲示板 4273～, 16058～ で議論がありました。なお、AMS 関係のパッケージを読み込む際に `psamsfonts` オプションを与えても回避できます (Thanks: しっぽ愛好家さん)。

[2016-11-19] 本家の `ltclass.dtx` 2004/01/28 v1.1g で修正されているのでコメントアウトしました。

```

%\let\ltx@ifnextchar\@ifnextchar
%\def\ProvidesFile#1{%
% \begingroup
% \catcode'\ 10 %
% \ifnum \endlinechar<256 %
% \ifnum \endlinechar>\m@ne
% \catcode\endlinechar 10 %
% \fi
% \fi
% \makeoother\/%
% \makeoother\&%
% \ltx@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

14 初期設定

■いろいろな語

```

\prepartname
\postpartname \newcommand{\prepartname}{\if@english Part~\else 第\fi}
\prechaptername \newcommand{\postpartname}{\if@english\else 部\fi}
\postchaptername \newcommand{\prechaptername}{\if@english Chapter~\else 第\fi}
\postsectionname \newcommand{\postchaptername}{\if@english\else 章\fi}
\presectionname
\postsectionname

```

```

\newcommand{\presectionname}{}% 第
\newcommand{\postsectionname}{}% 節

\contentsname
\listfigurename \newcommand{\contentsname}{\if@english Contents\else 目次\fi}
\listtablename \newcommand{\listfigurename}{\if@english List of Figures\else 図目次\fi}
\newcommand{\listtablename}{\if@english List of Tables\else 表目次\fi}

\refname
\bibname \newcommand{\refname}{\if@english References\else 参考文献\fi}
\indexname \newcommand{\bibname}{\if@english Bibliography\else 参考文献\fi}
\newcommand{\indexname}{\if@english Index\else 索引\fi}

\figurename
\tablename <!\jpf>\newcommand{\figurename}{\if@english Fig.~\else 図\fi}
<jspf>\newcommand{\figurename}{Fig.~}
<!\jpf>\newcommand{\tablename}{\if@english Table~\else 表\fi}
<jspf>\newcommand{\tablename}{Table~}

\appendixname
\abstractname % \newcommand{\appendixname}{\if@english Appendix~\else 付録\fi}
\newcommand{\appendixname}{\if@english \else 付録\fi}
<!book>\newcommand{\abstractname}{\if@english Abstract\else 概要\fi}

```

■今日の日付 L^AT_EX で処理した日付を出力します。jarticle 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

\newif\if 西暦 \西暦 true
\def\西暦{\西暦 true}
\def\和暦{\西暦 false}
\newcount\heisei \heisei\year \advance\heisei-1988\relax
\def\today{%
  \if@english
    \ifcase\month\or
      January\or February\or March\or April\or May\or June\or
      July\or August\or September\or October\or November\or December\fi
    \space\number\day, \number\year
  \else
    \if 西暦
      \number\year 年
      \number\month 月
      \number\day 日
    \else
      平成\number\heisei 年
      \number\month 月
      \number\day 日
    \fi
  \fi
}

```

```

\fi
\fi}

```

■ハイフネーション例外 T_EX 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）

```

\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。

```

<article>\if@slide \pagestyle{empty} \else \pagestyle{plain} \fi
<book>\if@report \pagestyle{plain} \else \pagestyle{headings} \fi
<report|kiyou>\pagestyle{plain}
<jspf>\pagestyle{headings}
\pagenumbering{arabic}
\if@twocolumn
  \twocolumn
  \sloppy
  \flushbottom
\else
  \onecolumn
  \raggedbottom
\fi
\if@slide
  \renewcommand\kanjifamilydefault{\gtdefault}
  \renewcommand\familydefault{\sfdefault}
  \raggedright
  \xkanjiskip=0.1em\relax
\fi

```

15 実験的コード

[2016-11-29] コミュニティ版 p_AT_EX で新設されたテスト用パッケージ（exppl2e パッケージ）が文書クラスより先に読み込まれていた場合は、jsclasses もテスト版として動作します。この処置は jsarticle, jsbook, jsreport にのみ行い、jspf と kiyou は除外しておきます。exppl2e パッケージが読みこまれていない場合は通常版として動作しますので、ここで終了します。

```

(*article|book|report)
\@ifpackageloaded{exppl2e}{\jsc@needspace@tchtrue}{\jsc@needspace@tchfalse}
\ifjsc@needspace@tch\else
  \expandafter\endinput
\fi

```

以下は実験的コードです。具体的には、2016/11/29 の exppl2e パッケージで説明されている \@gnewline のパッチを入れてあります。

```

\@gnewline
\def\@gnewline #1{%

```

```
\ifvmode
  \@nolnerr
\else
  \unskip \reserved@a {\reserved@a#1}\nobreak \hfil \break \hskip \z@
  \ignorespaces
\fi}
</article | book | report>
```

以上です。