

私家版マクロ集

奥村晴彦

2017/08/31

1 はじめに

これは『pL^AT_EX 2_ε 美文書作成入門』などの本を作ったときに使ったマクロを集めたものです。作っただけでテストもしていないものなども混じっている可能性があります。概して他の方にはあまり利用価値がないものです。

[2016-07-30] 従来は, `ascmac` → `okumacro` の順に読み込んだ場合にエラーが出ていましたが, 新しい仕様では, `ascmac` と `okumacro` の両方を読み込む場合には「後に読み込んだほう」が勝ちます。具体的には `\keytop`, `\return`, `screen` 環境, `shadebox` 環境があります。

2 数値集

- B5: 182 mm × 257 mm
- B4: 257 mm × 364 mm
- A5: 148 mm × 210 mm
- A4: 210 mm × 297 mm
- 1 in = 72.27 pt = 2.54 cm
- 1 Q = 0.25 mm = 0.711319 pt
- 1 pt = 1.405839 Q = 0.3514598 mm
- min10: 1 zw = 9.62216 pt = 3.3818 mm = 13.5272 Q, 1 zh = 9.16443 pt
- cmr10: 1 em = 10 pt, 1 ex = 4.30554 pt

3 雑マクロ

以下が実際のマクロ集です。

`(*okumacro)`

`\ruby` `\ruby{奥}{おく}``\ruby{村}{むら}`^{おくむら} で奥村となります。`\ruby{奥村}{おくむら}` でもかまいませんが, 奥と村の間で行分割ができなくなります。

和文ベースラインが 88 : 12 の位置にあることを仮定して支柱を入れました。

バグ : `1\ruby{奥}{おく}1` のような場合に `\xkanjiskip` が入りません。

[2003-02-09] ゴシック体のルビがゴシック体になるバグ?を修正しました。

[2003-05-29] 齋藤修三郎さんの OTF パッケージの expert オプションに対応しました。

[2003-09-01] オプション引数で幅が指定できるようにしました [2003-10-02 バグフィックス]。

[2010-03-14] ルビの直後で改行できなかったので、親字が和文と仮定して直後に `\kanjiskip` を入れました。

[2016-08-16] 縦組に対応。

[2016-08-16] 「前に `\kanjiskip` が入らない」「前後の欧文文字との間に `\xkanjiskip` が入らない」「後ろの禁則処理が効かない」問題を解決するために和文ゴースト処理を追加。

```
\providecommand{\rubyfamily}{}
\def\kanjistrut{\iftdir
  \vrule \@height0.5zw \@depth0.5zw \@width\z@
\else
  \vrule \@height0.88zw \@depth0.12zw \@width\z@
\fi}
\chardef\okumacro@zsp=\jis"2121\relax
\newcommand{\ruby}[3][0zw]{%
  \okumacro@zsp
  \kern-1zw\relax
  \dimen3=\f@size\p@
  \setbox1=\hbox{#2}%
  \setbox3=\hbox{\rubyfamily\fontsize{0.5\dimen3}{0pt}\selectfont #3}%
  \dimen1=#1\relax
  \ifdim\dimen1=\z@
    \ifdim\wd1>\wd3 \dimen1=\wd1 \else \dimen1=\wd3 \fi
  \else
    \ifdim\dimen1<\wd1
      \dimen1=\wd1
    \fi
  \fi
  \hbox{%
    \kanjiskip=0pt plus 2fil
    \xkanjiskip=0pt plus 2fil
    \vbox{%
      \hbox to \dimen1{\rubyfamily
        \fontsize{0.5\dimen3}{0pt}\selectfont \kanjistrut
        \ifdim\dimen1<\wd3
          \hss\unhbox3\hss
        \else
          \hfil\unhbox3\hfil
        \fi
      }%
      \nointerlineskip
      \hbox to \dimen1{\kanjistrut\hfil\unhbox1\hfil}}}%
  \kern-1zw\relax\okumacro@zsp}
```

`\Ruby` ルビが両側にルビの半文字分ずつはみ出してもいい場合に使います。ただ、これを連続して

使うと、ルビが重なり合ってしまいます。

これはあまり存在意義がないのでとりあえず消しておきます。

```
% \newcommand{\Ruby}[2]{%
%   \leavevmode
%   \setbox0=\hbox{#1}%
%   \mytempdima=\f@size\p@
%   \setbox1=\hbox{\fontsize{0.5\mytempdima}{0pt}\selectfont #2}%
%   \dimen0=\wd1 \advance \dimen0 by -0.5\mytempdima
%   \ifdim\wd0>\dimen0 \dimen0=\wd0 \fi
%   \hbox{%
%     \kanjiskip=0pt plus 2fil
%     \xkanjiskip=0pt plus 2fil
%     \vbox{%
%       \hbox to \dimen0{\normalfont\rubyfamily
%         \fontsize{0.5\mytempdima}{0pt}\selectfont \kanjistrut\hss#2\hss}%
%       \nointerlineskip
%       \hbox to \dimen0{\kanjistrut\hfil#1\hfil}}}}}
```

`\kenten` 圏点を振るコマンドです。

[2016-07-30] ルビと圏点の高さを合わせるつもりでしたが、間違って上シフト量を 0.63zw としていましたので、正しい値 0.75zw に直しました。

[2017-03-11] 「前後の欧文文字との間に `\xkanjiskip` が入らない」「後ろの禁則処理が効かない」問題を解決するために和文ゴースト処理を追加。

```
\def\kenten#1{%
  \okumacro@zsp
  \kern-1zw\relax
  \setbox1=\hbox to \z@{\cdot\hss}%
  \ht1=.63zw
  \@kenten#1\end\relax
  \kern-1zw\relax\okumacro@zsp}
\def\@kenten#1#2{%
  \ifx#1\end \let\next=\relax \else
    \raise.75zw\copy1\nobreak #1\ifx#2\end\else\hskip\kanjiskip\relax\fi
    \let\next=\@kenten
  \fi\next#2}
```

`\kintou` 簡単な均等割りコマンドです。

```
\newcommand{\kintou}[2]{%
  \leavevmode
  \hbox to #1{%
    \kanjiskip=0pt plus 1fil minus 1fil
    \xkanjiskip=\kanjiskip
    #2}}
```

`\曜` 今日の曜日を出力します。

```
\newcommand{\曜}{\@tempcnta=\year \@tempcntb=\month
  \ifnum \@tempcntb<3
    \advance \@tempcnta by -1
```

```

\advance \@tempcntb by 12
\fi
\multiply \@tempcntb by 13
\advance \@tempcntb by 8
\divide \@tempcntb by 5
\advance \@tempcntb by \@tempcnta
\divide \@tempcnta by 4
\advance \@tempcntb by \@tempcnta
\divide \@tempcnta by 25
\advance \@tempcntb by -\@tempcnta
\divide \@tempcnta by 4
\advance \@tempcntb by \@tempcnta
\advance \@tempcntb by \day
\@tempcnta=\@tempcntb
\divide \@tempcntb by 7
\multiply \@tempcntb by 7
\advance \@tempcnta by -\@tempcntb
\ifcase \@tempcnta 日\or 月\or 火\or 水\or 木\or 金\or 土\fi}}

```

\挨拶 時候の挨拶を出力します。

```

\def\挨拶{\noindent 拝啓\hspace{1zw}\ifcase\month\or
  厳寒\or 春寒\or 早春\or 陽春\or 新緑\or 向暑\or
  猛暑\or 残暑\or 初秋\or 仲秋\or 晩秋\or 初冬\fi
  の候、ますますご清栄のこととお喜び申し上げます。}

```

\mytt \texttt では | ! @ * - + = [] : ; " ' < > ? , . / はそのまま出せますが、 ! ? . : の後は余分な空きが入ります。また、 # \$ & _ { } は \ で逃げれば使えますが、そうしても _ { } は正しい cmtt フォントになりません。さらに、 ~ ^ は \ を付けても後続の文字のアクセントのようになってしまいます (\~{ }, ^{ } のようにすれば何とかありますが)。ここで定義する \mytt マクロを使えば、 { } は \ で逃げれば正しい cmtt フォントになり、 # & _ ~ ^ は \ で逃げなくてもそのまま出力できます。\$ もそのまま出力できるようにしていたのですが、数式モードは使えた方が便利なので、外しました。また、 \\ で円印が出ます。/, \\, ., @, - の後ろで改行できるようにしてあります。{\mytt ...} のようにして使います (\mytt{...} ではありません)。スペースは若干伸縮するようにしました。 \mbox{\mytt ...} のようにして使えば、和文・欧文間のグルーの自動挿入はしません。

```

\def\myallowbreak{\@ifnextchar\egroup{\penalty\exhyphenpenalty}{}}
\begingroup
\catcode'\-=\active
\catcode'\.=\active
\catcode'\/= \active
\gdef\myth@k{%
\catcode'\-=\active \def-\{\char'\-\myallowbreak}%
\catcode'\.=\active \def.\{\char'\.\myallowbreak}%
\catcode'\/= \active \def/{\char'\/\myallowbreak}%
}
\endgroup
\def\mytt{\ttfamily

```

```

\catcode'\#=12
\catcode'\&=12
\catcode'\^=12
\catcode'\~=12
\catcode'\_ =12
\chardef\{='\{
\chardef\}='}\}
\chardef\_='\_
% \def\{Y\llap=\penalty\exhyphenpenalty{}}% ¥記号
\def\{\char"5C\penalty\exhyphenpenalty{}}% バックスラッシュ
% \spaceskip=.5em plus.25em minus.15em
\spaceskip=.32em plus.16em minus.09em
\frenchspacing
\noautoxspacing
\mytth@k}

```

これ以外の記号は次のようにして出力できます。

```

\texttt{\char"20}    □ 見える空白
\texttt{\char"5C}    \ バックスペース
\texttt{\char"0B}    ↑ 上向き矢印
\texttt{\char"0C}    ↓ 下向き矢印
\texttt{\char"0D}    ' 左右対称なシングルクォート

```

`\yen` `\yen` は円印, `\BS` はバックスラッシュです。

```

\BS \DeclareRobustCommand{\yen}{\oalign{Y\cr\hss=\hss}}
\providecommand{\BS}{\texttt{\char"5C}}

```

`\bslash` バックスラッシュは doc パッケージなどでの定義に従っておきます。

```

{\catcode'\|=z@ \catcode'\=12\gdef\bslash{\}}

```

`\asciibar` 真ん中が少し途切れた縦棒 | です。| に対応する ASCII 文字です。

```

\newcommand{\asciibar}{%
\makebox[0.525em]{\rule[-0.2ex]{0.15ex}{0.7ex}\kern-0.15ex
\rule[0.8ex]{0.15ex}{0.7ex}}}

```

`\---` 倍角のダークです。昔は `\rule[0.33zh]{2zw}{.03zh}` としていましたが、ちょっと低すぎるようです。和文フォントのベースラインは 88 : 12 のところにあるものが多いので、0.38zw のところに中心線が来るほうがいいようです。太さもちょっとだけ太くしました。

```

% \def\---{\rule[0.36zw]{2zw}{0.04zw}}

```

さらによく考えたら `\scalebox{2}[1]{-}` のほうがよさそうなので、`graphics` または `graphicx` パッケージを読み込んだ場合はこちらを使うようにしました。

```

% \def\---{\@ifundefined{scalebox}{\rule[0.36zw]{2zw}{0.04zw}}{\scalebox{2}[1]{-}}}

```

さらにいろいろなご意見をいただき、次のように直しました。

```

\def\---{-\kern-.5zw-\kern-.5zw-}

```

上のものは全角マイナス二つでしたが、全角ダッシュ二つでもいいようにしました。

```
\def\--{-\kern-.5zw-\kern-.5zw}
```

[2016-12-04] pL^AT_EX で通るコードが upL^AT_EX で通らなくなることを防ぐために、upL^AT_EX の場合には U+FF0D, U+2014 で定義したのも別途用意しました。

```
\ifx\ucs\@undefined\else
  \begingroup
    \kansujichar1="FF0D
    \kansujichar2="2014
    \expandafter\expandafter\expandafter\gdef
      \expandafter\csname\expandafter\kansuji\expandafter1%
        \expandafter\endcsname\kansuji1{-\kern-.5zw-\kern-.5zw}
    \expandafter\expandafter\expandafter\gdef
      \expandafter\csname\expandafter\kansuji\expandafter2%
        \expandafter\endcsname\kansuji2{-\kern-.5zw-\kern-.5zw}
  \endgroup
\fi
```

4 キートップ, ○数字

`\removept` #1 に与えた `dimen` レジスタの `pt` を消した数値を #2 のマクロに定義します。

```
\newcommand{\removept}[2]{%
  \edef#2{\expandafter\@removePT\the#1}
  {\catcode'p=12 \catcode't=12 \gdef\@removePT#1pt{#1}}
  % しかし latex.ltx で次のように定義されていました。
  % これをそのまま使ったらよかった。
  % \begingroup
  %   \catcode'P=12
  %   \catcode'T=12
  %   \lowercase{
  %     \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}
  %   \expandafter\endgroup\x
  % \def\strip@pt{\expandafter\rem@pt\the}
```

`\keytop` これを使ってキートップ記号を作ります。

[2016-07-30] `\keytop` は pL^AT_EX の `ascmac` パッケージ (`tascmac.sty`) でも定義されていますが, `ascmac` の定義では周囲と同じファミリーのままになっていて, `okumacro` では中の文字をサンセリフ体に変更しています。両方のパッケージを読み込む場合は, 後に読み込んだほうの定義が勝ちます。 `okumacro` が勝つ場合には常にサンセリフ体に変更するキートップ記号を出力しますし, `ascmac` が勝つ場合は周囲と同じファミリーになります。

また, 新たに `\okukeytop` という命令を追加します。これは `ascmac` と干渉せず, 常にサンセリフ体に変更するキートップ記号を出力できます。




```
\newcommand{\okukeytop}[1]{\leavevmode\kern1pt
  \setbox1=\hbox{\normalfont\sffamily []}%
  \dimen1=\ht1
  \removept{\dimen1}{\tempC}%
```


```

\setbox0=\hbox{\normalfont\sffamily #1}%
\dp0=0pt
\dimen0=\wd0
\ifdim\dimen0<\ht1 \dimen0=\ht1 \fi
\advance \dimen0 by 5pt
\removept{\dimen0}{\tempA}%
\divide \dimen0 by 2
\removept{\dimen0}{\tempB}%
\unitlength=1pt
\advance \dimen1 by 2pt
\hbox{\vrule height\dimen1 depth2pt width0pt}%
\begin{picture}(\tempA,\tempC)
  \put(0,0){\line(0,1){\tempC}}
  \put(\tempA,0){\line(0,1){\tempC}}
  \put(\tempB,0){\oval(\tempA,4)[b]}
  \put(\tempB,\tempC){\oval(\tempA,4)[t]}
  \put(\tempB,0){\makebox(0,0)[b]{\box0}}
\end{picture}\kern1pt}}
\expandafter\ifx\csname ver@tascmac.sty\endcsname\relax
  \newcommand{\keytop}{\okukeytop}
\else
  \renewcommand{\keytop}{\okukeytop}
\fi

```

`\return` リターンキーの記号です。

[2016-07-30] `\return` は p $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ の `ascmac` パッケージ (`tascmac.sty`) でも定義されていますが、`okumacro` がリターンのキートップ記号を出力するのに対し、`ascmac` ではリターンの矢印文字  だけを出す命令になっています。両方のパッケージを読み込む場合は、後に読み込んだほうの定義が勝ちます。`okumacro` が勝つ場合には  を出力しますし、`ascmac` が勝つ場合は  を出力します。

また、新たに `\okureturn` という命令を追加します。これは `ascmac` と干渉せず、常にリターンのキートップ記号  を出力できます。

```

\def\RETMARK{{\unitlength=1pt
  \setbox0=\hbox{\normalfont\ttfamily []}%
  \dimen0=\ht0
  \removept{\dimen0}{\tempA}%
  \advance \dimen0 by -1pt
  \removept{\dimen0}{\tempB}%
  \begin{picture}(\tempA,\tempA)
    \put(\tempA,1){\line(0,1){\tempB}}
    \put(\tempA,1){\vector(-1,0){\tempA}}
  \end{picture}}}
\newcommand{\okureturn}{\okukeytop{\RETMARK}}
\expandafter\ifx\csname ver@tascmac.sty\endcsname\relax
  \newcommand{\return}{\okureturn}
\else
  \renewcommand{\return}{\okureturn}

```

`\fi`

`\upkey` 上下右左の矢印キーです。

```
\downkey \newcommand{\upkey}{\raisebox{-2pt}{%
\setlength{\unitlength}{1pt}%
\rightkey \thinlines
\leftkey \begin{picture}(14,9)
\put(7,2){\vector(0,1){8}}
\put(1,2){\line(0,1){7}}
\put(13,2){\line(0,1){7}}
\put(7,9){\oval(12,4)[t]}
\put(7,2){\oval(12,4)[b]}
\end{picture}%
}}
\newcommand{\downkey}{\raisebox{-2pt}{%
\setlength{\unitlength}{1pt}%
\thinlines
\begin{picture}(14,9)
\put(7,10){\vector(0,-1){8}}
\put(1,2){\line(0,1){7}}
\put(13,2){\line(0,1){7}}
\put(7,9){\oval(12,4)[t]}
\put(7,2){\oval(12,4)[b]}
\end{picture}%
}}
\newcommand{\rightkey}{\raisebox{-2pt}{%
\setlength{\unitlength}{1pt}%
\thinlines
\begin{picture}(14,9)
\put(3,6){\vector(1,0){8}}
\put(1,2){\line(0,1){7}}
\put(13,2){\line(0,1){7}}
\put(7,9){\oval(12,4)[t]}
\put(7,2){\oval(12,4)[b]}
\end{picture}%
}}
\newcommand{\leftkey}{\raisebox{-2pt}{%
\setlength{\unitlength}{1pt}%
\thinlines
\begin{picture}(14,9)
\put(11,6){\vector(-1,0){8}}
\put(1,2){\line(0,1){7}}
\put(13,2){\line(0,1){7}}
\put(7,9){\oval(12,4)[t]}
\put(7,2){\oval(12,4)[b]}
\end{picture}%
}}
}}
```

`\MARU` 丸で囲んだ文字です。

[2003-01-24] graphicx のとき少し見映えが良くなるようにしました。

```
% \newcommand{\MARU}[1]{%
% \oalign{\hfil#1/\hfil\crr\raise.167ex\hbox{\mathhexbox20D}}}}
\newcommand{\MARU}[1]{%
  \oalign{\hfil
    \@ifundefined{scalebox}{%
      #1/\hfil\crr\raise.167ex}{\raise.1zw\hbox{\scalebox{0.8}{#1/}}\hfil\crr\raise.1zw}
    \hbox{\fontencoding{OMS}\fontfamily{cmsy}\fontseries{m}\fontshape{n}\selectfont
      \char"0D}}}}
```

その他のロゴです。

```
% \newcommand{\MuTeX}{M\kern-.15em\raise.2ex\hbox{u}\kern-.32em\TeX}
\newcommand{\PiC}{P\kern-.12em\lower.5ex\hbox{I}\kern-.075emC}
\newcommand{\PiCTeX}{\PiC\kern-.11em\TeX}
\newcommand{\JTeX}{\leavevmode\lower.5ex\hbox{J}\kern-.17em\TeX}
\newcommand{\JLaTeX}{\leavevmode\lower.5ex\hbox{\rm J}\kern-.17em\LaTeX}
\DeclareRobustCommand{\JBibTeX}{\mbox{%
  \lower.5ex\hbox{J}\ifx\family\cmr\kern-.15em\else\kern-.05em\fi
  \BibTeX}$}
\newcommand{\pTeXsT}{p\kern-.21em\TeX\kern-.10em s\kern-.21em T}
% インプレスの新 TeX の仮称
\newcommand{\iTeX}{\accent'27\i\TeX}
% 桜井さんの新 TeX のロゴ (仮)
\newcommand{\MlTeX}{M\kern-.1667em\lower.5ex\hbox{L}\kern-.25em\TeX}
```

METAFONT のロゴを出す方法はいろいろあります。まず古い方法で、manfnt を使うものです。

```
\font\manual=manfnt
\font\manLarge=manfnt scaled \magstep 2
\font\manHuge=manfnt scaled \magstep 5
\newcommand{\MF}{%
  \ifx\currsize\small
    {\manual hijk}\-{\manual lmnj}% 9pt
  \else
    \ifx\currsize\footnotesize
      \ifdim \fontdimen\@ne\font>\z@
        {\manual \char92\char93\char94\char95efg\char94}% 8pt slanted
      \else
        {\manual opqr}\-{\manual stuq}% 8pt
      \fi
    \else
      \ifx\currsize\Large
        {\manLarge META}\-{\manLarge FONT}% mag2 (14pt)
      \else
        \ifx\currsize\Huge
          {\manHuge META}\-{\manHuge FONT}% mag5 (25pt)
        \else % \normalsize
          \ifdim \fontdimen\@ne\font>\z@
```

```

        {\manual 89:;<=>:}% 10pt slanted
    \else
        {\manual META}\-{\manual FONT}% 10pt
    \fi
    \fi
    \fi
    \fi
\fi}

```

次は logo を使う方法です。

```

\font\logoTen=logo10
\font\logoEight=logo8
\font\logoLarge=logo10 scaled \magstep 2
\font\logoHuge=logo10 scaled \magstep 5
\newcommand{\MF}{%
    \ifx\@currsize\Large
        {\logoLarge META}\-{\logoLarge FONT}%
    \else
        \ifx\@currsize\Huge
            {\logoHuge META}\-{\logoHuge FONT}%
        \else
            \ifx\@currsize\footnotesize
                {\logoEight META}\-{\logoEight FONT}%
            \else
                {\logoTen META}\-{\logoTen FONT}%
            \fi
        \fi
    \fi}

```

L^AT_EX 2_ε の方法としては、次のような ulogo.fd を作っておきます。

```

\DeclareFontFamily{U}{logo}{%
\DeclareFontShape{U}{logo}{m}{n}{%
    <5><6><7><8>logo8%
    <9>logo9%
    <10><10.95><12><14.4><17.28><20.74><24.88>logo10}{%
\DeclareFontShape{U}{logo}{m}{sl}{%
    <5><6><7><8><9>logosl9%
    <10><10.95><12><14.4><17.28><20.74><24.88>logosl10}{%
\DeclareFontShape{U}{logo}{bx}{n}{%
    <5><6><7><8><9>%
    <10><10.95><12><14.4><17.28><20.74><24.88>logobf10}{%
\DeclareFontShape{U}{logo}{m}{it}
    {<->ssub*logo/m/sl}{%

```

これで次のようにします。

```

\DeclareRobustCommand{\MF}{\fontfamily{logo}\selectfont META}\-%
    {\fontfamily{logo}\selectfont FONT}}

```

```
\DeclareRobustCommand{\MP}{\fontfamily{logo}\selectfont META}\-%
{\fontfamily{logo}\selectfont POST}}
```

しかしよく調べてみたら同様なものが `mflogo.sty`, `uologo.fd` として配布されていたので、そちらを使うことにします。

5 いろいろな環境

`dangerous` 参考事項をメモ風に付ける環境です。鉛筆記号を使うようにしました。

```
%\newcommand{\dangerouslabel}[1]{%
% \setlength{\fboxsep}{1pt}%
% \fbox{\footnotesize\bfseries #1}}
\newenvironment{dangerous}{%
\vspace{3pt}%
\begin{list}{\ding{"2E}}{%
% \let\makelabel=\dangerouslabel
\setlength{\leftmargin}{3zw}%
\setlength{\itemindent}{0pt}%
\setlength{\labelwidth}{2zw}%
\setlength{\labelsep}{6pt}%
\setlength{\rightmargin}{0pt}%
\setlength{\itemsep}{0.5\baselineskip}%
\setlength{\parsep}{0pt}%
\setlength{\topsep}{0.5\baselineskip}%
\setlength{\partopsep}{0pt}%
\setlength{\listparindent}{0zw}%
}\small}\end{list}}
```

`namelist` `\begin{namelist}{xxxx} \item[...] ... \end{namelist}` で見出しを `xxxx` の幅にしたリストを作ります。

```
% \newcommand{\namelistlabel}[1]{\#1\ \dotfill\ }
\newcommand{\namelistlabel}[1]{\hspace{1zw}\#1\hfil}
\newenvironment{namelist}[1]{%
\begin{list}{}%
\let\makelabel=\namelistlabel
\settowidth{\labelwidth}{\#1}%
\addtolength{\labelwidth}{2zw}%
\setlength{\leftmargin}{\labelwidth}%
\setlength{\labelsep}{0pt}%
}\end{list}}
```

`mybibliography` `thebibliography` とほぼ同じ参考文献リスト環境ですが、章(節)見出しを出力せず、カウンタをリセットしません。`thebibliography` と同じ `enumiv` というカウンタを使っていますので、必要に応じて

```
\setcounter{enumiv}{0}
```

として初期化してください。

```

\newenvironment{mybibliography}[1]{%
  \list{\@biblabel{\@arabic{c@enumiv}}}%
    {\settowidth\labelwidth{\@biblabel{#1}}%
      \leftmargin\labelwidth
      \advance\leftmargin\labelsep
      \@openbib@code
      \@nmblisttrue\def\@listctr{enumiv}}%
  \let\p@enumiv\@empty
  \renewcommand\theenumiv{\@arabic{c@enumiv}}%
\slippy
\clubpenalty4000
\@clubpenalty\clubpenalty
\widowpenalty4000%
\sfcodes'\.\@m}
{\def\@noitemerr
  {\@latex@warning{Empty 'mybibliography' environment}}%
\endlist}

```

FRAME 枠です。

```

\newenvironment{FRAME}{\begin{trivlist}\item[]
  \hrule width\linewidth
  \hbox to \linewidth\bgroup
    \advance\linewidth by -30pt
    \hsize=\linewidth
    \vrule\hfill
    \vbox\bgroup
      \vskip15pt
      \def\thempfootnote{\arabic{mpfootnote}}% 脚注を算用数字に
      \begin{minipage}{\linewidth}}{%
        \end{minipage}\vskip15pt
      \egroup\hfill\vrule
    \egroup\hrule width\linewidth
  \end{trivlist}}
\newcommand{\SHUTTEN}[1]{%
  \par\nobreak\removeatlastskip\rightline{\small#1}\medbreak}

```

screen スクリーン風の環境です。元は `ascmac.sty` にあったのだと思いますが、かなり改変したかもしれません。

`\begin{screen}[0.75]` のようにオプションの数値を与えると横幅を 0.75 倍するようにしてみました。倍率が 1 以外の場合は `graphicx` パッケージが必要です。画面出力例がコラム幅を超える場合の対処法として有効かもしれません。ただ、`screen` 環境の最初に `[` が来る場合は誤動作しますので、`\relax` か `{}` を入れて誤魔化してください。

[2016-07-30] 元になっている `pLATEX` の `ascmac` パッケージ (`tascmac.sty`) の `screen` 環境は、オプション引数の数値で角の丸みの大きさを整数値 (0-8) で指定しますが、`okumacro` パッケージでは横の倍率を指定するという違いがあります。両方のパッケージを読み込む場合は、後に読み込んだほうの定義が勝ちます。`screen` 環境は、`okumacro` が勝つ場合には横の倍率を指定できるスクリーン風の環境になりますし、`ascmac` が勝つ場合は角の丸みを指

定できるスクリーン風の環境になります。

[2016-10-08] 画面出力例がコラム幅を超える場合の対処法として横倍率指定のオプションを追加しましたが、同じ枠内に改行可能な文章も含まれる場合に枠の右端から離れた不自然な位置（横倍率を縮めない場合に端にあっていた位置）で改行していました。これを修正しました。

また、新たに okuscreen 環境を追加します。これは ascmac と干渉せず、常に横の倍率を指定できるスクリーン風の環境になります。

```
\expandafter\ifx\cscname ver@tascmac.sty\endcsname\relax
  \newdimen\@scw
  \newbox\scb@x
\fi
\newenvironment{okuscreen}[1][1]%
{
\def\screensc@le{#1}\@scw=\linewidth \advance \@scw by -20pt
\dimen1=#1\p@\relax
\@tempcnta=\dimen1\relax
\@tempcntb=65536\relax
\divide\@scw by \@tempcnta
\multiply\@scw by \@tempcntb
\setbox\scb@x=\hbox\bgroup\begin{minipage}[b]{\@scw}}%
  % または \setbox\scb@x=\vbox\bgroup\advance \linewidth by -20pt \relax
\end{minipage}\egroup
\dimen0=\linewidth \removept{\dimen0}{\hsPT}%
\divide \dimen0 by 2 \removept{\dimen0}{\hhsPT}%
\dimen0=\ht\scb@x \advance \dimen0 by 10pt
\removept{\dimen0}{\htxPT}%
\begin{flushleft}
  \vspace{6.5pt}%
  \begin{picture}(\hsPT,\htxPT)
    \put(\hhsPT,\htxPT){\oval(\hsPT,10)[t]}
    \put(0,0){\line(0,1){\htxPT}}
    \put(\hsPT,0){\line(0,1){\htxPT}}
    \put(\hhsPT,0){\oval(\hsPT,10)[b]}
    \put(10,5){\ifdim\screensc@le pt=1pt \box\scb@x
      \else\scalebox{\screensc@le}[1]{\box\scb@x}\fi}
  \end{picture}%
\end{flushleft}}
\let\screen\okuscreen \let\endscreen\endokuscreen
```

EXAMPLE 出力例です。ずいぶん前に作ったので、再検討の必要があります。

```
\newdimen\EXAMPLEWIDTH
\newenvironment{EXAMPLE}{\begin{flushleft}\EXAMPLEWIDTH=\linewidth
}{\end{flushleft}}
\newenvironment{IN}{%
  \makebox[35pt][l]{\small\fbbox{入力}}%
  \advance \linewidth by -35pt
  \begin{minipage}[t]{\linewidth}}{%
  \end{minipage}\[\6pt]}
```

```

\newenvironment{OUT}{%
  \makebox[35pt][l]{\small\fbbox{出力}}%
  \advance \linewidth by -35pt
  \begin{minipage}[t]{\linewidth}}{%
  \end{minipage}}
\newcommand{\INEX}{\medbreak\noindent
  \framebox[\columnwidth][l]{\small\hskip2zw 入力}\nobreak
  \begin{quote}\small}
\newcommand{\OUTEX}{\end{quote}}
\newcommand{\columnwidth}[1]{\small\hskip2zw 出力}\nobreak
  \begin{quotation}\small}
\newcommand{\ENDEX}{\end{quotation}\bigbreak}
\newcommand{\ENDEXC}[1]{\end{quotation}\nobreak
  \begin{flushright}\footnotesize#1\end{flushright}\bigbreak}

```

6 雑

`\whichpage` `\whichpage{参照名}` で、そのページが同じページにあれば何も出力せず、そうでなければ (次ページ) (前ページ) (123 ページ) のような形式で場所を出力します。

```

\def\whichpage#1{%
  \expandafter\ifx\csname r@#1\endcsname\relax
  \else
    \@tempcnta\pageref{#1}\relax
    \advance\@tempcnta-\c@page
    \ifnum\@tempcnta=0
      \else
        \ifnum\@tempcnta=1 (次ページ)
        \else
          \ifnum\@tempcnta=-1 (前ページ)
          \else (\pageref{#1}ページ)
          \fi
        \fi
      \fi
    \fi
  \fi
}

```

`migiake` ごく簡単な右空けコマンドです。使っていません。

```

% \newenvironment{migiake}[1]{%
%   \setlength{\rightskip}{#1}\samepage}{\par}

```

`\rightfig` もう少しましな右空けです。#1 は空け幅、#2 は行数、#3 は図や表です。

```

% \def\rightfig#1#2#3{\noindent
%   \dimen0=\baselineskip \multiply \dimen0 by #2
%   \advance\dimen0 by -\baselineskip \advance\dimen0 by 2ex
%   \hangindent=-#1 \hangafter=-#2
%   \raise 2ex \hbox to 0pt{\smash{\vtop to \dimen0{\vfil
%     \vbox{\hfill \hbox to #1{ \ \hfill #3}}\vfil
%     }}\hss}\hskip\parindent\ignorespaces}

```

漢字の幅を tt フォント 2 文字分にします。使っていません。

```
% \newcommand{\kanjitt}[1]{\texttt{\kanjiskip0.083595em\relax#1}}
```

shadebox 影付きの箱です。ascmac.sty から取りました。

[2016-07-30] 元になっている pL^AT_EX の ascmac パッケージ (tascmac.sty) と衝突しないように、新しい仕様では、ascmac と okumacro の両方を読み込む場合には「後に読み込んだほう」が勝ちます。

また、新たに okushadebox 環境を追加します。これは ascmac と干渉せず、常に okumacro 流の定義になります。

```
\expandafter\ifx\cscname ver@tascmac.sty\endcscname\relax
\newdimen\shaderule \shaderule 5\p@
\fi
\def\okushadebox{\dimen0\linewidth \advance\dimen0-20\p@
\advance\dimen0-2\fbboxrule \advance\dimen0-\shaderule
\setbox\@tempboxa=\hbox\bgroup\minipage{\dimen0}}
\def\endokushadebox{\endminipage\egroup\dimen0=10\p@ \advance\dimen0-\fbboxrule
\setbox\@tempboxa=\hbox{\kern\dimen0\unhbox\@tempboxa\kern\dimen0}%
\setbox0=\vbox{\hrule\@height \fbboxrule
\hbox{\vrule\@width \fbboxrule \hskip-\fbboxrule
\vbox{\vskip\fbboxsep \box\@tempboxa\vskip\fbboxsep}%
\hskip-\fbboxrule\vrule \@width \fbboxrule}\hrule \@height \fbboxrule}%
\dimen0=\ht0 \advance\dimen0-\shaderule
\hbox{\hbox to \shaderule{\copy0\hss}\kern \z@
\vrule\@width\wd0\@height\z@\@depth\shaderule\hskip-\shaderule
\vrule\@width\shaderule\@height\dimen0}}
\let\shadebox\okushadebox \let\endshadebox\endokushadebox
```

sankou 参考 (dangerous bend 風) の環境です。

```
\def\sankoumark{{\small ★}}
\newenvironment{sankou}{\vspace{3pt}%
\begin{list}{}{%
\setlength{\leftmargin}{3zw}%
\setlength{\itemindent}{0pt}%
\setlength{\labelwidth}{2zw}%
\setlength{\labelsep}{6pt}%
\setlength{\rightmargin}{0pt}%
\setlength{\itemsep}{0pt}%
\setlength{\parsep}{0pt}%
\setlength{\topsep}{0.5\baselineskip}%
\setlength{\partopsep}{0pt}%
\setlength{\listparindent}{0zw}%
}\item[\sankoumark]\small}\end{list}}
```

toi 問いの環境です。

```
\def\toimark{{\small\bf 問}}
\newenvironment{toi}{\vspace{3pt}%
\begin{list}{}{%
\setlength{\leftmargin}{3zw}%
```

```

\setlength{\itemindent}{0pt}%
\setlength{\labelwidth}{2zw}%
\setlength{\labelsep}{6pt}%
\setlength{\rightmargin}{0pt}%
\setlength{\itemsep}{0pt}%
\setlength{\parsep}{3pt plus 1pt minus 1pt}%
\setlength{\topsep}{3pt plus 1pt minus 1pt}%
\setlength{\partopsep}{3pt plus 6pt minus 1pt}%
\setlength{\listparindent}{0zw}%
\item[\toimark]\small\end{list}}

```

英語用の行送りに切り替えるコマンドですが、使っていません。

```

% \def\english{%
% \ifx\@currsize\normalsize \fontsize{10}{12pt}\selectfont\else
% \ifx\@currsize\small \fontsize{9}{11pt}\selectfont\else
% \ifx\@currsize\footnotesize \fontsize{8}{9.5pt}\selectfont\fi\fi\fi}
% \newcommand{\shima}{dvi\out\slash dvi\prt\index{dviout/dviprt}}
\providecommand{\eps}{\varepsilon}
% \newcommand{\zip}{\bar{\kern0pt}\relax}
% \def\mathpalette#1#2{\mathchoice{#1\displaystyle{#2}}%
% {#1\textstyle{#2}}{#1\scriptstyle{#2}}{#1\scriptscriptstyle{#2}}}%
% ≤, ≥, = の上下に点
\def\LEQQ{\mathrel{\mathpalette\gl@align<}}
\def\GEQQ{\mathrel{\mathpalette\gl@align>}}
\def\gl@align#1#2{\lower.6ex\vbox{\baselineskip\z@skip\lineskip\z@
\ialign{\$m\th#1\hfil##\hfil$\crrc#2\crrc=\crrc}}}%
\def\APPROX{\mathrel{\mathpalette\ap@align{\smash.}}}%
\def\ap@align#1#2{\lower.2ex\vbox{\baselineskip\z@skip\lineskip\z@
\def\finsh{\ht\z@.2ex \dp\z@.2ex \box\z@}%
\ialign{\$m\th#1\hfil##\hfil$\crrc#2\crrc=\crrc#2\crrc}}}%
%% Fractions like 1/2 by DEK (TUGboat, 6(1):36, TeXbook (Exercise 11.6))
\def\FRAC#1#2{\leavevmode\kern.1em
\raise.5ex\hbox{\the\scriptfont0 #1}\kern-.1em
/\kern-.15em\lower.25ex\hbox{\the\scriptfont0 #2}}
\newcommand{\hk}{\hspace{\kanjiskip}}
\newcommand{\hx}{\hspace{\xkanjiskip}}
\newcommand{\3}{\penalty300 }
\newcommand{\5}{\penalty50 } % \penalty\exhyphenpenalty

```

\> 全角コンマの直後に半角文字が来た場合、全角コンマの後に挿入して、半角空白と同じだけの伸縮を持たせるためのグルー挿入コマンドです。使っていません。

```

% \newcommand{\>}{\relax}
% \ifmmode\mskip\medmuskip\else\hskip0em plus0.167em minus0.111em\fi}

```

\ANGLE Angle brackets です。

```

\newcommand{\ANGLE}[1]{%
\langle$\kern0pt\nobreak{#1}\kern0pt\nobreak$\rangle$}

```

分数関係です。amstex パッケージでもほぼ同じものが定義されていますが、amstex パッ

ページの `\cfrac` では `\mathstrut` ではなく `\strut` が使われているため、和文スタイルで行送りが大きい場合には、ちょっと間延びしてしまいます。

```
% \providecommand{\tfrac}[2]{\textstyle\frac{#1}{#2}}
% \providecommand{\dfrac}[2]{\displaystyle\frac{#1}{#2}}
% \providecommand{\cfrac}[2]{\dfrac{\mathstrut #1}{#2}}
```

`\dalembert` D'Alembertian (`\Box` より少々大きい四角) です。

```
% \newcommand{\dalembert}{%
% \setlength{\unitlength}{0.1em}%
% \begin{picture}(8,8)
% \thinlines
% \put(1,0){\framebox(6,6){}}
% \end{picture}}
```

`\kanjibox` 漢字の大きさの箱です。使っていません。

```
% \newcommand{\kanjibox}{\setlength{\unitlength}{0.1091175zw}%
% \begin{picture}(9.62216,7.77588)\thinlines
% \put(0,-1.38855){\framebox(9.16443,9.16443){}}\end{picture}}
```

Poor person's blackboard bold です*1。

`\R`

```
\C % \def\R{{\rm I\kern-.17em R}} % (-.25em in the above literature)
% \def\C{{\rm\kern.24em\vrule width.02em height1.4ex depth-.05ex\kern-.26em C}}
```

`\株`

```
\newcommand{\株}{\hspace{\kanjiskip}\< (株) \<\hspace{\kanjiskip}}
```

`\CVS` CVS 用。

```
% \def\CVS$#1: #2 ${\expandafter\def\csname CVS#1\endcsname{#2}}

\CVS$Revision: 1.7 $
\CVS$Date: 2000/06/09 05:55:02 $
この文書のバージョンは\CVSRevision 版である。
```

のように使います。

[2003-02-16..17] `js*` クラスで数式の前後のスペースが若干変わってしまいます。とりあえず `amsmath` の `equation(*)`, `align(*)` 環境の類だけここで直しておきます。このあたりはまた見直します。

```
\@ifundefined{widebaselines}{%
\let\endmathdisplaysave=\endmathdisplay
\def\endmathdisplay{\widebaselines\endmathdisplaysave}}
\let\start@alignsave=\start@align
\def\start@align{\vspace{4pt}\start@alignsave}
```

以上で終わりです。

```
</okumacro>
\endinput
```

*1 Robert Messer, "Blackboard Bold." *TUGboat* 9(1):19–20, 1988.