

# 私家版マクロ集

奥村晴彦

2021/07/01

## 1 はじめに

これは『 $\text{\LaTeX}$  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。

[2002-12-19] いろいろなものに収録していただく際にライセンスを明確にする必要が生じてきました。アスキーのものが最近は (modified) BSD ライセンスになっていますので、私のものもそれに準じて (modified) BSD とすることにします。

[2016-07-30] 従来は, `ascmac`  $\rightarrow$  `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  $\times$  257 mm
- B4: 257 mm  $\times$  364 mm
- A5: 148 mm  $\times$  210 mm
- A4: 210 mm  $\times$  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。

1 `\*okumacro`

`\ruby \ruby{奥}{おく}\ruby{村}{むら}` <sup>おくむら</sup> 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を、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。

```
2 \providecommand{\rubyfamily}{  
3 \def\kanjistrut{\iftdir  
4   \vrule \@height0.5zw \@depth0.5zw \@width\z@  
5 \else  
6   \vrule \@height0.88zw \@depth0.12zw \@width\z@  
7 \fi}  
8 \chardef\okumacro@zsp=\jis"2121\relax  
9 \newcommand{\ruby}[3][0zw]{%  
10  \okumacro@zsp  
11  \kern-1zw\relax  
12  \dimen3=\f@size\p@  
13  \setbox1=\hbox{#2}%  
14  \setbox3=\hbox{\rubyfamily\fontsize{0.5\dimen3}{0pt}\selectfont #3}%  
15  \dimen1=#1\relax  
16  \ifdim\dimen1=\z@  
17    \ifdim\wd1>\wd3 \dimen1=\wd1 \else \dimen1=\wd3 \fi  
18  \else  
19    \ifdim\dimen1<\wd1  
20      \dimen1=\wd1  
21    \fi  
22  \fi  
23  \hbox{%  
24    \kanjiskip=0pt plus 2fil  
25    \xkanjiskip=0pt plus 2fil  
26    \vbox{%  
27      \hbox to \dimen1{\rubyfamily  
28        \fontsize{0.5\dimen3}{0pt}\selectfont \kanjistrut  
29        \ifdim\dimen1<\wd3  
30          \hss\unhbox3\hss  
31        \else  
32          \hfil\unhbox3\hfil  
33        \fi  
34      }%  
35    \nointerlineskip
```

```

36      \hbox to \dimen1{\kanjistrut\hfil\unhbox1\hfil}}}%
37      \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。

```

38 % \newcommand{\Ruby}[2]{%
39 %   \leavevmode
40 %   \setbox0=\hbox{#1}%
41 %   \mytempdima=\f@size\p@
42 %   \setbox1=\hbox{\fontsize{0.5\mytempdima}{0pt}\selectfont #2}%
43 %   \dimen0=\wd1 \advance \dimen0 by -0.5\mytempdima
44 %   \ifdim\wd0>\dimen0 \dimen0=\wd0 \fi
45 %   \hbox{%
46 %     \kanjiskip=0pt plus 2fil
47 %     \xkanjiskip=0pt plus 2fil
48 %     \vbox{%
49 %       \hbox to \dimen0{\normalfont\rubyfamily
50 %         \fontsize{0.5\mytempdima}{0pt}\selectfont \kanjistrut\hss#2\hss}%
51 %       \nointerlineskip
52 %       \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。

[2017-10-04] もともと \kenten では

```
\setbox1=\hbox to \z@{\cdot\hss}%
```

としていましたが、2017 年 9 月の pTeX(p3.7.2) に対応するため

```
\setbox1=\hbox to\z@{\hbox to 1zw{\hss \cdot\hss}\hss}%
```

に変更しました。これで、p3.7.1 以前・p3.7.2 以降どちらも OK です。

```

53 \def\kenten#1{%
54   \okumacro@zsp
55   \kern-1zw\relax
56   \setbox1=\hbox to\z@{\hbox to 1zw{\hss \cdot\hss}\hss}%
57   \ht1=.63zw
58   \@kenten#1\end\relax
59   \kern-1zw\relax\okumacro@zsp}
60 \def\@kenten#1#2{%
61   \ifx#1\end \let\next=\relax \else
62     \raise.75zw\copy1\nobreak #1\ifx#2\end\else\hskip\kanjiskip\relax\fi
63     \let\next=\@kenten
64   \fi\next#2}

```

`\kintou` 簡単な均等割りコマンドです。

```
65 \newcommand{\kintou}[2]{%
66   \leavevmode
67   \hbox to #1{%
68     \kanjiskip=0pt plus 1fil minus 1fil
69     \xkanjiskip=\kanjiskip
70     #2}}
```

`\曜` 今日の曜日を出力します。

```
71 \newcommand{\曜}{\@tempcnta=\year \@tempcntb=\month
72   \ifnum \@tempcntb<3
73     \advance \@tempcnta by -1
74     \advance \@tempcntb by 12
75   \fi
76   \multiply \@tempcntb by 13
77   \advance \@tempcntb by 8
78   \divide \@tempcntb by 5
79   \advance \@tempcntb by \@tempcnta
80   \divide \@tempcnta by 4
81   \advance \@tempcntb by \@tempcnta
82   \divide \@tempcnta by 25
83   \advance \@tempcntb by -\@tempcnta
84   \divide \@tempcnta by 4
85   \advance \@tempcntb by \@tempcnta
86   \advance \@tempcntb by \day
87   \@tempcnta=\@tempcntb
88   \divide \@tempcntb by 7
89   \multiply \@tempcntb by 7
90   \advance \@tempcnta by -\@tempcntb
91   \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。

```
92 \def\挨拶{\noindent 拝啓 \hspace{1zw}\ifcase\month\or
93   厳寒\or 春寒\or 早春\or 陽春\or 新緑\or 向暑\or
94   猛暑\or 残暑\or 初秋\or 仲秋\or 晩秋\or 初冬\fi
95   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。

```

96 \def\myallowbreak{\@ifnextchar\egroup{\{\penalty\exhyphenpenalty\}}\}
97 \begingroup
98   \catcode'\-=\active
99   \catcode'\.=\active
100  \catcode'\/= \active
101  \gdef\mytth@@k{%
102    \catcode'\-=\active \def-\{\char'\-\myallowbreak\}%
103    \catcode'\.=\active \def.\{\char'\.\myallowbreak\}%
104    \catcode'\/= \active \def/{\char'\/\myallowbreak\}%
105  }
106 \endgroup
107 \def\mytt{\ttfamily
108   \catcode'\#=12
109   \catcode'\&=12
110   \catcode'\^=12
111   \catcode'\~=12
112   \catcode'\_ =12
113   \chardef\{=' \{
114   \chardef\}=' \}
115   \chardef\_ =' \_
116 % \def\{\Yllap=\penalty\exhyphenpenalty\}}% ¥記号
117 \def\{\char"5C\penalty\exhyphenpenalty\}}% バックスラッシュ
118 % \spaceskip=.5em plus.25em minus.15em
119 \spaceskip=.32em plus.16em minus.09em
120 \frenchspacing
121 \noautoxspacing
122 \mytth@@k}

```

これ以外の記号は次のようにして出力できます。

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

`\yen \yen` は円印, `\BS` はバックスラッシュです。

```

\BS 123 \DeclareRobustCommand{\yen}{\ooalign{\Y\cr\hss=\hss}}
124 \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。

```

125 {\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。

```

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

```

`\--` 倍角のダッシュです。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よいようです。太さもちょっとだけ太くしました。

```
129 % \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。

```
130 % \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。

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

上のものは全角マイナス二つでしたが、全角ダシ二つでもよいようにしました。

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

[2016-12-04] p<sub>La</sub>T<sub>E</sub>X で通るコードが up<sub>La</sub>T<sub>E</sub>X で通らなくなることを防ぐために、up<sub>La</sub>T<sub>E</sub>X の場合には U+FF0D、U+2014 で定義したものも別途用意しました。

```
133 \ifnum\jis"2121="3000 % upTeX check
134   \begingroup
135     \kansujichar1="FF0D
136     \kansujichar2="2014
137     \expandafter\expandafter\expandafter\gdef
138     \expandafter\csname\expandafter\kansuji\expandafter1%
139     \expandafter\endcsname\kansuji1{\kern-.5zw—\kern-.5zw}
140     \expandafter\expandafter\expandafter\gdef
141     \expandafter\csname\expandafter\kansuji\expandafter2%
142     \expandafter\endcsname\kansuji2{\kern-.5zw—\kern-.5zw}
143   \endgroup
144 \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。

```
145 \newcommand{\removept}[2]{%
146   \edef#2{\expandafter\@removePT\the#1}}
147 {\catcode'p=12 \catcode't=12 \gdef\@removePT#1pt{#1}}
148 % しかし latex.ltx で次のように定義されていました。
149 % これをそのまま使ったらよかった。
150 % \begingroup
151 %   \catcode'P=12
152 %   \catcode'T=12
153 %   \lowercase{
154 %     \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}
155 %   \expandafter\endgroup\x
156 % \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  $\text{pLATEX}$  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。


```

157 \newcommand{\okukeytop}[1]{\leavevmode\kern1pt
158   \setbox1=\hbox{\normalfont\sffamily []}%
159   \dimen1=\ht1
160   \removept{\dimen1}{\tempC}%
161   \setbox0=\hbox{\normalfont\sffamily #1}%
162   \dp0=0pt
163   \dimen0=\wd0
164   \ifdim\dimen0<\ht1 \dimen0=\ht1 \fi
165   \advance \dimen0 by 5pt
166   \removept{\dimen0}{\tempA}%
167   \divide \dimen0 by 2
168   \removept{\dimen0}{\tempB}%
169   \unitlength=1pt
170   \advance \dimen1 by 2pt
171   \hbox{\vrule height\dimen1 depth2pt width0pt}%
172   \begin{picture}(\tempA,\tempC)
173     \put(0,0){\line(0,1){\tempC}}
174     \put(\tempA,0){\line(0,1){\tempC}}
175     \put(\tempB,0){\oval(\tempA,4)[b]}
176     \put(\tempB,\tempC){\oval(\tempA,4)[t]}
177     \put(\tempB,0){\makebox(0,0)[b]{\box0}}
178   \end{picture}\kern1pt}}
179 \expandafter\ifx\csname ver@tascmac.sty\endcsname\relax
180   \newcommand{\keytop}{\okukeytop}
181 \else
182   \renewcommand{\keytop}{\okukeytop}
183 \fi

```

`\return` リターンキーの記号です。

[2016-07-30] `\return` は  $\text{pLATEX}$  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。

```

184 \def\RETMARK{\unitlength=1pt
185   \setbox0=\hbox{\normalfont\ttfamily []}%
186   \dimen0=\ht0

```

```

187 \removept{\dimen0}{\tempA}%
188 \advance \dimen0 by -1pt
189 \removept{\dimen0}{\tempB}%
190 \begin{picture}(\tempA,\tempA)
191   \put(\tempA,1){\line(0,1){\tempB}}
192   \put(\tempA,1){\vector(-1,0){\tempA}}
193 \end{picture}}
194 \newcommand{\okureturn}{\okukeytop{\RETMARK}}
195 \expandafter\ifx\csname ver@tasclac.sty\endcsname\relax
196   \newcommand{\return}{\okureturn}
197 \else
198   \renewcommand{\return}{\okureturn}
199 \fi

```

\upkey 上下右左の矢印キーです。

```

\downkey 200 \newcommand{\upkey}{\raisebox{-2pt}{%
\rightkey 201   \setlength{\unitlength}{1pt}%
202   \thinlines
\leftkey 203   \begin{picture}(14,9)
204     \put(7,2){\vector(0,1){8}}
205     \put(1,2){\line(0,1){7}}
206     \put(13,2){\line(0,1){7}}
207     \put(7,9){\oval(12,4)[t]}
208     \put(7,2){\oval(12,4)[b]}
209   \end{picture}%
210 }}
211 \newcommand{\downkey}{\raisebox{-2pt}{%
212   \setlength{\unitlength}{1pt}%
213   \thinlines
214   \begin{picture}(14,9)
215     \put(7,10){\vector(0,-1){8}}
216     \put(1,2){\line(0,1){7}}
217     \put(13,2){\line(0,1){7}}
218     \put(7,9){\oval(12,4)[t]}
219     \put(7,2){\oval(12,4)[b]}
220   \end{picture}%
221 }}
222 \newcommand{\rightkey}{\raisebox{-2pt}{%
223   \setlength{\unitlength}{1pt}%
224   \thinlines
225   \begin{picture}(14,9)
226     \put(3,6){\vector(1,0){8}}
227     \put(1,2){\line(0,1){7}}
228     \put(13,2){\line(0,1){7}}
229     \put(7,9){\oval(12,4)[t]}
230     \put(7,2){\oval(12,4)[b]}
231   \end{picture}%
232 }}
233 \newcommand{\leftkey}{\raisebox{-2pt}{%

```



```

234 \setlength{\unitlength}{1pt}%
235 \thinlines
236 \begin{picture}(14,9)
237 \put(11,6){\vector(-1,0){8}}
238 \put(1,2){\line(0,1){7}}
239 \put(13,2){\line(0,1){7}}
240 \put(7,9){\oval(12,4)[t]}
241 \put(7,2){\oval(12,4)[b]}
242 \end{picture}%
243 }}

```

\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。

```

244 % \newcommand{\MARU}[1]{%
245 % \oalign{\hfil#1/\hfil\crrc\raise.167ex\hbox{\mathhexbox20D}}}%
246 \newcommand{\MARU}[1]{%
247 \oalign{\hfil
248 \ifundefined{scalebox}{%
249 #1/\hfil\crrc\raise.167ex}{\raise.1zw\hbox{\scalebox{0.8}{#1/}}}\hfil\crrc\raise.1zw
250 \hbox{\fontencoding{OMS}\fontfamily{cmsy}\fontseries{m}\fontshape{n}\selectfont
251 \char"OD}}}%

```

その他のロゴです。

```

252 % \newcommand{\MuTeX}{M\kern-.15em\raise.2ex\hbox{u}\kern-.32em\TeX}
253 \newcommand{\PiC}{P\kern-.12em\lower.5ex\hbox{I}\kern-.075emC}
254 \newcommand{\PiCTeX}{\PiC\kern-.11em\TeX}
255 \newcommand{\JTeX}{\leavevmode\lower.5ex\hbox{J}\kern-.17em\TeX}
256 \newcommand{\JLaTeX}{\leavevmode\lower.5ex\hbox{\rm J}\kern-.17em\LaTeX}
257 \DeclareRobustCommand{\JBibTeX}{\m@mth\mbox{%
258 \lower.5ex\hbox{J}\kern-.05em\BibTeX}$}
259 \newcommand{\pTeXsT}{p\kern-.21em\TeX\kern-.10em s\kern-.21em T}
260 % インプレスの新 TeX の仮称
261 \newcommand{\iTeX}{\accent'27\i\TeX}
262 % 桜井さんの新 TeX のロゴ (仮)
263 \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<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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`, `ulogo.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` (*env.*) 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。

```

264 %\newcommand{\dangerouslabel}[1]{\%
265 % \setlength{\fboxsep}{1pt}%
266 % \fbox{\footnotesize\bfseries #1}}
267 \newenvironment{dangerous}{%
268 \vspace{3pt}%
269 \begin{list}{\ding{"2E}}{%
270 % \let\makelabel=\dangerouslabel
271 \setlength{\leftmargin}{3zw}%
272 \setlength{\itemindent}{0pt}%
273 \setlength{\labelwidth}{2zw}%
274 \setlength{\labelsep}{6pt}%
275 \setlength{\rightmargin}{0pt}%
276 \setlength{\itemsep}{0.5\baselineskip}%
277 \setlength{\parsep}{0pt}%
278 \setlength{\topsep}{0.5\baselineskip}%
279 \setlength{\partopsep}{0pt}%
280 \setlength{\listparindent}{0zw}%
281 }\small}{\end{list}}

```

`namelist` (*env.*) `\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。

```

282 % \newcommand{\namelistlabel}[1]{\#1\ \dotfill\ }
283 \newcommand{\namelistlabel}[1]{\hspace{1zw}\#1\hfil}
284 \newenvironment{namelist}[1]{%
285 \begin{list}{\}%
286 \let\makelabel=\namelistlabel
287 \settowidth{\labelwidth}{\#1}%
288 \addtolength{\labelwidth}{2zw}%

```

```

289 \setlength{\leftmargin}{\labelwidth}%
290 \setlength{\labelsep}{0pt}%
291 }{\end{list}}

```

`mybibliography` (*env.*) `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。

```

292 \newenvironment{mybibliography}[1]{%
293 \list{\@biblabel{\@arabic{c@enumiv}}}%
294 {\settowidth\labelwidth{\@biblabel{#1}}%
295 \leftmargin\labelwidth
296 \advance\leftmargin\labelsep
297 \@openbib@code
298 \@nmblisttrue\def\@listctr{enumiv}%
299 \let\p@enumiv\@empty
300 \renewcommand\theenumiv{\@arabic{c@enumiv}}}%
301 \sloppy
302 \clubpenalty4000
303 \@clubpenalty\clubpenalty
304 \widowpenalty4000%
305 \sfcode'\.\@m}
306 {\def\@noitemerr
307 {\@latex@warning{Empty 'mybibliography' environment}}}%
308 \endlist}

```

`FRAME` (*env.*) 枠です。

```

309 \newenvironment{FRAME}{\begin{trivlist}\item[]
310 \hrule width\linewidth
311 \hbox to \linewidth\bggroup
312 \advance\linewidth by -30pt
313 \hsize=\linewidth
314 \vrule\hfill
315 \vbox\bggroup
316 \vskip15pt
317 \def\thempfootnote{\arabic{mpfootnote}}% 脚注を算用数字に
318 \begin{minipage}{\linewidth}}{%
319 \end{minipage}\vskip15pt
320 \egroup\hfill\vrule
321 \egroup\hrule width\linewidth
322 \end{trivlist}}
323 \newcommand{\SHUTTEN}[1]{%
324 \par\nobreak\removeatlastskip\rightline{\small#1}\medbreak}

```

`screen` (*env.*) 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 p<sub>La</sub>T<sub>E</sub>X 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たっていた位置) 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。

[2018-06-11] 2017/10/04 版までは、枠の中身の深さを考慮していなかったもので、表 (`tabular`) や深さのある数式を囲むと枠からはみ出していました。この問題を修正しましたが、なるべく既存文書のレイアウトが変わらないように、通常のテキストの深さ分だけ引いておきました (! と印をつけた部分)。

```

325 \expandafter\ifx\csname ver@tascmac.sty\endcsname\relax
326   \newdimen\@scw
327   \newbox\scb@x
328 \fi
329 \newenvironment{okuscreen}[1][1]%
330 {\def\screensc@le{#1}\@scw=\linewidth \advance \@scw by -20pt
331   \dimen1=#1\p@\relax
332   \@tempcnta=\dimen1\relax
333   \@tempcntb=65536\relax
334   \divide\@scw by \@tempcnta
335   \multiply\@scw by \@tempcntb
336   \setbox\scb@x=\hbox\bgroup\begin{minipage}[b]{\@scw}}%
337   % または \setbox\scb@x=\vbox\bgroup\advance \linewidth by -20pt \relax
338 {\end{minipage}\egroup
339   \dimen0=\linewidth \removept{\dimen0}{\hsPT}%
340   \divide \dimen0 by 2 \removept{\dimen0}{\hhsPT}%
341   \dimen0=\dp\scb@x \advance \dimen0 by 5pt
342   \advance \dimen0 by -.5\cdp %% !
343   \removept{\dimen0}{\hdxPT}%
344   \advance \dimen0 by \ht\scb@x \advance \dimen0 by 5pt
345   \advance \dimen0 by -.5\cdp %% !
346   \removept{\dimen0}{\htxPT}%
347   \begin{flushleft}
348     \vspace{6.5pt}%

```

```

349 \begin{picture}(\hsPT,\htxPT)
350 \put(\hhsPT,\htxPT){\oval(\hsPT,10)[t]}
351 \put(0,0){\line(0,1){\htxPT}}
352 \put(\hsPT,0){\line(0,1){\htxPT}}
353 \put(\hhsPT,0){\oval(\hsPT,10)[b]}
354 \put(10,\hdxPT){\ifdim\screen@le pt=1pt \box\scb@x
355 \else\scalebox{\screen@le}[1]{\box\scb@x}\fi}
356 \end{picture}%
357 \end{flushleft}}
358 \let\screen\okuscreen \let\endscreen\endokuscreen

```

EXAMPLE (env.) 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。

```

359 \newdimen\EXAMPLEWIDTH
360 \newenvironment{EXAMPLE}{\begin{flushleft}\EXAMPLEWIDTH=\linewidth
361 }{\end{flushleft}}
362 \newenvironment{IN}{%
363 \makebox[35pt][l]{\small\fbbox{入力}}%
364 \advance \linewidth by -35pt
365 \begin{minipage}[t]{\linewidth}}{%
366 \end{minipage}}\[6pt]}
367 \newenvironment{OUT}{%
368 \makebox[35pt][l]{\small\fbbox{出力}}%
369 \advance \linewidth by -35pt
370 \begin{minipage}[t]{\linewidth}}{%
371 \end{minipage}}
372 \newcommand{\INEX}{\medbreak\noindent
373 \framebox[\columnwidth][l]{\small\hskip2zw 入力}\nobreak
374 \begin{quote}\small}
375 \newcommand{\OUTEX}{\end{quote}
376 \framebox[\columnwidth][l]{\small\hskip2zw 出力}\nobreak
377 \begin{quotation}\small}
378 \newcommand{\ENDEX}{\end{quotation}\bigbreak}
379 \newcommand{\ENDEXC}[1]{\end{quotation}\nobreak
380 \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。

```

381 \def\whichpage#1{%
382 \expandafter\ifx\csname r@#1\endcsname\relax
383 \else
384 \@tempcnta\pageref{#1}\relax
385 \advance\@tempcnta-\c@page
386 \ifnum\@tempcnta=0
387 \else
388 \ifnum\@tempcnta=1 (次ページ)

```

```

389      \else
390      \ifnum\@tempcnta=-1 (前ページ)
391      \else (\pageref{#1}-ページ)
392      \fi
393      \fi
394      \fi
395  \fi}

```

`migiake (env.)` ごく簡単な右空けコマンドです。使っていません。

```

396 % \newenvironment{migiake}[1]{%
397 %   \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。

```

398 % \def\rightfig#1#2#3{\noindent
399 %   \dimen0=\baselineskip \multiply \dimen0 by #2
400 %   \advance\dimen0 by -\baselineskip \advance\dimen0 by 2ex
401 %   \hangindent=-#1 \hangafter=-#2
402 %   \raise 2ex \hbox to Opt{\smash{\vtop to \dimen0{\vfil
403 %     \vbox{\hfill \hbox to #1{ \ \hfill #3}}\vfil
404 %   }}\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。

```

405 % \newcommand{\kanjitt}[1]{\texttt{\kanjiskip0.083595em\relax#1}}

```

`shadebox (env.)` 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 p<sub>A</sub>T<sub>E</sub>X 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。

```

406 \expandafter\ifx\csname ver@tascmac.sty\endcsname\relax
407   \newdimen\shaderule \shaderule 5\p@
408 \fi
409 \def\okushadebox{\dimen0\linewidth \advance\dimen0-20\p@
410   \advance\dimen0-2\fbxrule \advance\dimen0-\shaderule
411   \setbox\@tempboxa=\hbox\bgroup\minipage{\dimen0}}
412 \def\endokushadebox{\endminipage\egroup\dimen0=10\p@ \advance\dimen0-\fbxrule
413   \setbox0=\hbox{\kern\dimen0\unhbox\@tempboxa\kern\dimen0}%
414   \setbox0=\vbox{\hrule\@height \fbxrule
415     \hbox{\vrule\@width \fbxrule \hskip-\fbxrule
416       \vbox{\vskip\fbxsep \box\@tempboxa\vskip\fbxsep}%
417       \hskip-\fbxrule\vrule \@width \fbxrule}\hrule \@height \fbxrule}%
418   \dimen0=\ht0 \advance\dimen0-\shaderule
419   \hbox{\hbox to \shaderule{\copy0\hss}\kern \z@
420     \vrule\@width\wd0\@height\z@\@depth\shaderule\hskip-\shaderule
421     \vrule\@width\shaderule\@height\dimen0}}
422 \let\shadebox\okushadebox \let\endshadebox\endokushadebox

```

sankou (env.) 参考 (dangerous bend 風) の環境です。

```

423 \def\sankoumark{{\small ★}}
424 \newenvironment{sankou}{\vspace{3pt}%
425   \begin{list}{}{%
426     \setlength{\leftmargin}{3zw}%
427     \setlength{\itemindent}{0pt}%
428     \setlength{\labelwidth}{2zw}%
429     \setlength{\labelsep}{6pt}%
430     \setlength{\rightmargin}{0pt}%
431     \setlength{\itemsep}{0pt}%
432     \setlength{\parsep}{0pt}%
433     \setlength{\topsep}{0.5\baselineskip}%
434     \setlength{\partopsep}{0pt}%
435     \setlength{\listparindent}{0zw}%
436   }\item[\sankoumark]\small}{\end{list}}

```

toi (env.) 問いの環境です。

```

437 \def\toimark{{\small\bf 【問】}}
438 \newenvironment{toi}{\vspace{3pt}%
439   \begin{list}{}{%
440     \setlength{\leftmargin}{3zw}%
441     \setlength{\itemindent}{0pt}%
442     \setlength{\labelwidth}{2zw}%
443     \setlength{\labelsep}{6pt}%
444     \setlength{\rightmargin}{0pt}%
445     \setlength{\itemsep}{0pt}%
446     \setlength{\parsep}{3pt plus 1pt minus 1pt}%
447     \setlength{\topsep}{3pt plus 1pt minus 1pt}%
448     \setlength{\partopsep}{3pt plus 6pt minus 1pt}%
449     \setlength{\listparindent}{0zw}%
450   }\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。

```

451 % \def\english{%
452 %   \ifx\@currsize\normalsize \fontsize{10}{12pt}\selectfont\else
453 %   \ifx\@currsize\small \fontsize{9}{11pt}\selectfont\else
454 %   \ifx\@currsize\footnotesize \fontsize{8}{9.5pt}\selectfont\fi\fi\fi}
455 % \newcommand{\shima}{dvi\out\slash dvi\prt\index{dviout/dviprt}}
456 \providecommand{\eps}{\varepsilon}
457 % \newcommand{\zip}{\kern0pt\relax}
458 % \def\mathpalette#1#2{\mathchoice{#1\displaystyle{#2}}%
459 %   {#1\textstyle{#2}}{#1\scriptstyle{#2}}{#1\scriptscriptstyle{#2}}}
460 % ≤, ≥, = の上下に点
461 \def\LEQQ{\mathrel{\mathpalette\gl@align<}}
462 \def\GEQQ{\mathrel{\mathpalette\gl@align>}}
463 \def\gl@align#1#2{\lower.6ex\vbox{\baselineskip\z@skip\lineskip\z@
464   \ialign{$\m@th#1\hfil##\hfil$\crcr#2\crcr=\crcr}}
465 \def\APPROX{\mathrel{\mathpalette\ap@align{\smash.}}}
466 \def\ap@align#1#2{\lower.2ex\vbox{\baselineskip\z@skip\lineskip\z@

```



```

467 \def\finism@sh{\ht\z@.2ex \dp\z@.2ex \box\z@}%
468 \ialign{$\m@th#1\hfil##\hfil$\crr#2\crr=\crr#2\crr}}
469 %%% Fractions like 1/2 by DEK (TUGboat, 6(1):36, TeXbook (Exercise 11.6))
470 \def\FRAC#1#2{\leavevmode\kern.1em
471 \raise.5ex\hbox{\the\scriptfont0 #1}\kern-.1em
472 /\kern-.15em\lower.25ex\hbox{\the\scriptfont0 #2}}
473 \newcommand{\hk}{\hspace{\kanjiskip}}
474 \newcommand{\hx}{\hspace{\xkanjiskip}}
475 \newcommand{\3}{\penalty300 }
476 \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。

```

477 % \newcommand{\>}{\relax
478 % \ifmode\mskip\medmuskip\else\hskip0em plus0.167em minus0.111em\fi}

```

\ANGLE Angle brackets です。

```

479 \newcommand{\ANGLE}[1]{%
480 $\langle$\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 パッケージの \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。

```

481 % \providecommand{\tfrac}[2]{\textstyle\frac{#1}{#2}}
482 % \providecommand{\dfrac}[2]{\displaystyle\frac{#1}{#2}}
483 % \providecommand{\cfrac}[2]{\dfrac{\mathstrut #1}{#2}}

```

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

```

484 % \newcommand{\dalembert}{%
485 % \setlength{\unitlength}{0.1em}%
486 % \begin{picture}(8,8)
487 % \thinlines
488 % \put(1,0){\framebox(6,6){}}
489 % \end{picture}}

```

\kanjibox 漢字の大きさの箱です。使っていません。

```

490 % \newcommand{\kanjibox}{\setlength{\unitlength}{0.1091175zw}%
491 % \begin{picture}(9.62216,7.77588)\thinlines
492 % \put(0,-1.38855){\framebox(9.16443,9.16443){}}\end{picture}}

```

Poor person's blackboard bold です<sup>\*1</sup>。

\R

```

\C 493 % \def\R{{\rm I\kern-.17em R}} % (-.25em in the above literature)
494 % \def\C{{\rm\kern.24em\vrule width.02em height1.4ex depth-.05ex\kern-.26em C}}

```

---

<sup>\*1</sup> Robert Messer, "Blackboard Bold." *TUGboat* 9(1):19–20, 1988.

\株

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

\CVS CVS 用。

```
496 % \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。

[2021-07-01] okumacro が有ると xalignat 上部のアキが大きくなる現象 (forum:3029) が報告されました。js\* クラスのための修正だったはずですが、js\* クラスを使った場合でも「修正しない方がバランスが良い」という挙動だったため、削除しました。

```
497 %\@ifundefined{widebaselines}{\}{%
```

```
498 % \let\endmathdisplaysave=\endmathdisplay
```

```
499 % \def\endmathdisplay{\widebaselines\endmathdisplaysave}}
```

```
500 % \let\start@alignsave=\start@align
```

```
501 % \def\start@align{\vspace{4pt}\start@alignsave}
```

以上で終わりです。

```
502 </okumacro>
```

```
503 \endinput
```