LaTeX 覚書

目次

1　\author 及び \date の省略方法

```
\author{\textbackslash vspace{-1zh}}
\date{\textbackslash vspace{-1zh}}
```

と入力する.

2　長さの単位

pt	ポイント (1pt ≈ 0.35mm)
pc	パイカ (1pc = 12pt)
bp	big point(1bp ≈ 1in)
dd	didot point(1dd ≈ 1.07pt)
mm	ミリメートル (1mm ≈ 2.85pt)
cm	センチメートル (1cm ≈ 10mm)
in	インチ (1in = 25.4mm)
em	現在有効な書体の文字 M の幅
ex	現在有効な書体の文字 x の高さ
zw	現在有効な全角漢字の幅
zh	現在有効な全角漢字の高さ

ただし，zw,zh は pTEX, pLATEX, upLATEX などのみに定義されている．LuaTEX や XeTEX で使いたい場合は\zw,\zh と指定する.

3　新しい長さの定義

次のようにすることで新しい長さを定義できる.

```
\newlength{長さの名前}
\setlength{長さの名前}{初期値}
```

4　特殊文字の出力

出力	入力
#	\#
$	\$
%	\%
&	\&
~	\textasciitilde
_	_
^	\textasciicircum
\	\textbackslash(数式モード内では\backslash)
{	\{
}	\}

5　余白レイアウトの仕方

```
1    \usepackage{\layout}
2    \begin{document}
3      \layout
4      ...
5    \end{document}
```

でレイアウトを出力して余白を調整したい部分の名称を確認する. その後プリアンブル部で

```
1          \setlength{余白部分の名称}{長さ}
2          \setlength{余白部分の名称}{長さ}
3          $\vdots$
```

のように調整する.LaTeX は文字サイズや用紙サイズを変更する時，標準のサイズに合わ

せてから拡大・縮小を行い，設定したサイズに変更をする．よって，通常の cm,mm を使って長さを調節した場合，拡大・縮小時に本来の長さも拡大・縮小されてしまう．これを防ぐには cm,mm の代わりに truecm,truemm を使う．

6　複数の著者と所属をつける方法

\usepackage{authblk}でパッケージを読み込んで

```
\author[1]{hoge}
\author[2]{fuga}

\affil[1]{piyo}
\affil[2]{hogehoge}
```

7　段落の最初に字下げしない方法

字下げしたくない段落で\noindent と書く. すべての段落で字下げを行わないようにするにはプリアンブル部に\parindent = 0pt と書く.

8　minted で載せたソースコードがページをまたぐ場合

通常,minted で載せたソースコードはページを跨いでも適切に処置してくれるが以下のいずれかの条件を満たす時, ソースコードがページをはみ出す.

- listing 環境を使った時
- minted に bgcolor オプションを付加した時

minted で載せたソースコードが 1 ページに収まらない場合は mdfaramed 環境を使う. 次のような構造で使う.

```
\usepackage{mdframed}
% TeXLive2015 まで
```

```
\usepackage[cache=false]{minted}
% TeXLive2016 以降
\usepackage{minted}
\begin{document}
\begin{mdframed}
    \begin{minted}{c}
    ...
    \end{minted}
\end{mdframed}
\end{document}
```

mdframed 環境はデフォルトでボックスで囲まれているので minted の上下左右の線とかぶる場合がある. その場合は次のようなオプションをつけるとよい.

```
\begin{mdframed}[
    topline=false,
    bottomline=false,
    leftline=false,
    rightline=false]
\end{mdframed}
```

mdframed 環境を使う場合は listing 環境が使えずキャプションを通常どおりにつけることができない. その為,captionof 環境を使う. 使い方としては次の通りである.

```
\usepackage{caption}
...
\captionof{*}{*}
...
```

captionof 環境は caption パッケージの中にあるので読み込む.captionof 環境の第一引数にはキャプションの種類を記述する. メジャーなのは次の 3 つである. 第二引数にはキャプションにする文章を入力する.

表示	引数に取る文字列
図○	figure
表○	table
Listing ○	listing

9　目次の作り方及び目次に表示する節のレベルを設定する

```
\tableofcontents
```

表示はできるが\subsubsection などの小々節は目次に載らない. こういう時は次のコマンドをプリアンブル部に記述する.

```
\setcounter{tocdepth}{3}
```

tocdepth というのは目次に載せる節の深さを表しておりデフォルトで 2 である. よって\subsubsection 以下のレベルは目次に載らない. なので\setcounter コマンドで変数に入っている値を変更する. 値を増やせば細かく目次に載り, 逆に値を減らせば大雑把な目次になる.

10　目次にありもしない項目を加える

\addcontentsline{file}{type}{text}で section で生成している節以外に目次に項目を増やせる.

表 1　引数 file の取りうる値

toc	part や section,subsection などで生成される一般的な目次
figure	図の一覧目次
table	表の一覧目次

type

section なら本来の目次の section 並の字下げを行う.subsection も然り.
text
目次に追加したい項目を書く

11　条件分岐を行うマクロなど ifthen パッケージ

条件分岐, 命題, 繰り返しなどほとんどプログラミング言語に近づけたような制御マクロが詰まっている ifthen パッケージ`\usepackage{ifthen}`とすれば使える. 詳しい使い方はここの URL から `http://qiita.com/zr_tex8r/items/71ae46c9c4e8cb575073#_reference-9aee38fe27ee3682b28a`

12　LaTeX3 でデフォルトになるマクロ定義マクロが入った xparse パッケージ

LaTeX3 のチームが開発したオプション引数が複数指定できたりと本来のマクロ定義コマンドよりも自由度が高いマクロ定義コマンドが入った xparse パッケージ. 使い方は以下から `http://qiita.com/zr_tex8r/items/50168ad7087516c3e139`

13　マクロ内で verbatim 環境を使う方法

`\usepackage{fancybox}`で fancybox パッケージを読み込む. 以下のように使う.

```
    \NewDocumentEnvironment{myverbatim}{%
    \VerbatimEnvironment
    \begin{Verbatim}
}
{
    \end{Verbatim}
}%
```

このように最初に`\VerbatimEnvironment` を指定する必要がある.

14　ラベルからページ番号を参照する方法

\label{}などとすると\ref{}で表，図，数式などの番号を参照することができる．しかし，\pageref{}とすると，参照元のページ番号を出力することができる

15　input した TEX ファイルなどを拡大・縮小する方法

graphcs パッケージを読み込み次のように記述する．

```
\usepackage{graphcs}
\begin{document}
...
\scalebox{倍率}{\input{ファイル名}}
...
\end{document}
```

また，横幅や縦幅に合わせたい場合は\resizebox を使う．

```
% 文章の横幅に合わせる場合
\resizebox{\textwidth}{!}{\input{ファイル名}}
% 文章の縦幅に合わせる場合
\resizebox{!}{\textheight}{\input{ファイル名}}
```

16　組版におけるクォートのルールについて

LATEX などの組版処理システムにおいてはクォートで文章を囲む時にルールがある．シングルクォートで囲む時は始まりは`(バッククォート) で始まり，'(シングルクォート) で終わる．ダブルクォートで囲む時は始まりは``(バッククォート 2 つ) 始まり，''(シングルクォート 2 つ) で終わる．

禁則処理の関係上，和文をクォートで囲む場合は全角のダブルクォートを使う (31 節

参照).

17　ページを跨ぐ表の作り方

通常の表は tabular 環境と table 環境を使って作るが 1 ページに収まらないような大きな表は通常, ページを跨ぐことができない. この場合は longtable パッケージを読み込み longtable 環境を使う.

```
1       \usepackage{longtable}
2       \begin{document}
3       \begin{longtable}[表の位置指定]{行数指定}
4           ...
5       \end{longtable}
```

表 2　表の位置指定に使えるオプション

オプション	効果
l	左詰め
c	中央揃え
r	右詰め

行数指定は通常の tabular 環境と一緒である.

18　表の各セルの高さを調整する方法

全てセルを高くするには次のようにする.

```
1       \renewcommand{\arraystretch{拡大率}}
```

一部のセルだけ高くするには rule コマンドを使う.

```
1   \newlength{¥ myheight}
```

```
\setlength{¥ myheight}{1cm}
\newlength{¥ myheighta}
\setlength{¥ myheighta}{2cm}

\begin{tabular}{|c|c|}
\hline
default & default \\
\hline
\rule{0cm}{\myheight} 1cm の高さ & ああ \\
\hline
\rule{0cm}{\myheighta} 2cm の高さ & いい \\
\hline
default & default \\
\hline
\end{tabular}
```

このコードの実行結果は次のようになる.

default	default
1cm の高さ	ああ
2cm の高さ	いい
default	default

19　美しいボックスを描ける tcolorbox 環境

使うには tcolorbox パッケージを読み込む. 例などは `http://ftp.jaist.ac.jp/pub/CTAN/macros/latex/contrib/tcolorbox/tcolorbox.pdf` に載っている.

20　tcolorbox 環境で美しい表を作る

作るには tcolorbox,array,tabularx のパッケージを読み込む.tcolorbox 環境のオプションで次のように指定することで表を作ることができる.

```
\begin{tcolorbox}[tabularx*={...]
        ...
\end{tcolorbox}
```

オプション内の... に続くのは実際に tabularx で使える命令やオプションである. 例として, 同じディレクトリにある tcolortable.tex を参照するとよい.

21　マクロで@を使う意味

@はカテゴリーコードで 12 に割り振られているのでその他の文字として扱われる。つまり、普通の TEX の文書を作っていて出てくる文字ではないという扱いである。そのため、マクロなどの名前に@を使うのは重複を防止するためである。@のカテゴリーコードを変更するには\makeatletter と\makeatother で囲む。

22　マクロの行末に % を入れる理由

TEX のスタイルファイルなどを見てみると行末に % だけが記述されていることがある. これは TEX の空白のルール上の対策である. TEX の空白が無視されるかどうかは次のルールに従う.

1. 行頭と行末にある空白文字・タブは全て無視される。
2. コメントの直後の改行文字は無視される。
3. 空行（連続する改行文字）は『改段落』を表す。
4. （空行以外の）改行文字は空白文字になる。
   （ただし pTeX 系で、行末が和文文字の場合は改行文字は無視される。）
5. 1 つ以上の空白文字・タブの並びは単一の空白文字と見做される。
6. 制御語（名前が英字からなる制御綴）の直後の空白文字は無視される。

7. それ以外の空白文字が『意味のある空白文字』となる。

上記のルールのうちの 4 番の影響で空白が入ってしまう為，ルールの 2 番に従って対策をする．よって % を入れる．

23　section などのフォントを変える方法

section のフォントはデフォルトで欧文:サンセリフ体，和文:ゴシック体である．これを変えるには次の\headfont を再定義する．

```
\renewcommand{\headfont}{<和文のフォント><欧文のフォント>}
```

24　文章全体のフォントを変える方法

文章全体の欧文及び和文のフォントを変えるには次のコマンドを再定義する．

```
\renewcommand{\kanjifamilydefault}{\gtdefault}
\renewcommand{\familydefault}{\sfdefault}
```

マクロ名はフォント名 +family ではなく，フォント名 +default なので注意

25　TEX で標準入力及び標準出力を行う方法

次のコマンドで標準入力を行う．

```
\newcount \tmp
\read-1 to \tmp
```

次のコマンドで標準出力を行う

```
\message{何か入力してください}
```

26　beamer でノートを使う方法

PowerPoint のような発表者用のノートは beamer では次のように入力する．

```
\note{...}
```

また，ノートだけを表示したい場合はプリアンブルで

```
\documentclass[notes=only]{beamer}
```

という風にする．また，どちらも表示したい場合は

```
\documentclass[notes]{beamer}
```

とする．

27　minted でアセンブラのソースを載せる時の注意

アセンブラといえどアセンブラの種類は CPU の種類だけあると言っていいほど書き方があるので minted の設定を間違えないように気をつける．minted 環境の第一引数に asm と持ってきがちだがこれは X86 のアセンブラなので intel のアセンブラとは違うものとなっている．アセンブラごとの引数の違いは URL 先のそれぞれの Short Name を参照すること．

`http://pygments.org/docs/lexers/`

minted は pyments をベースにシンタックスハイライトしている．pytments は字句解析 (lexer) を行うツールなので引数のとり方によっては字句解析の結果，過多なシンタックスハイライトがついてる場合があるので邪魔だと思う場合は名前に lexer とついていないものを選ぶこと．因みに asm は字句解析を行う．

28　TEX で言語処理する第一歩

28.1　TEX で文字列置換

TEX で文字列置換を行うには\@tfor\ch:=hoge\do{}を使う．これは hoge という文字列を\@tfor で一文字ずつ見ていくというマクロである．\@tfor で回している間のカウンタ変数は\ch である．次に例を挙げる．次の例は x という文字を \x に置き換えるマクロである．

```
    \@tfor\ch:=#1\do{\ifthenelse{\equal{\ch}{x}}{\textbackslash x}{\ch}}%
```

29　ドットの長さで後尾を揃えてくれる \dotfill

\dotfill はドットの前の文字列の長さに合わせてドットを出力して後尾の位置を合わせてくれるマクロである．以下，例．

```
    おこ \dotfill レベル 1 \\
    激おこ \dotfill レベル 2 \\
    激おこぷんぷん丸 \dotfill レベル 3 \\
    ムカ着火ファイア \dotfill レベル 4 \\
    カム着火ｲﾝﾌｪﾙﾉｫｫｫｵｵｵｳ \dotfill レベル 5 \\
    げきオコスティックファイナリアリティぷんぷんドリーム \dotfill レベル 6
```

出力結果

おこ ..... レベル 1
激おこ ..... レベル 2
激おこぷんぷん丸 ..... レベル 3
ムカ着火ファイア ..... レベル 4
カム着火ｲﾝﾌｪﾙﾉｫｫｫｵｵｵｳ ..... レベル 5
げきオコスティックファイナリアリティぷんぷんドリーム ..... レベル 6

30　verbatim 環境内で自動改行を有効にする spverbatim 環境

verbatim 環境は入力したテキストをそのまま出力する為，自動改行されない．spverbatim 環境を使うと自動改行を有効にする．

31　LaTeX の自動改行のしくみ

LaTeX の自動改行は禁則処理に基づいて行われる．全角と半角の禁則処理は異なる．自動改行のしくみが分からないと直感的に理解することができない仕様がある．以下に事例を挙げる．

表 3　全角と半角の禁則処理概要

全角 or 半角	禁則処理概要
全角	全角括弧や全角クォートは分割不可
半角	空白以外全て分割不可

31.1　クォートがおかしな位置に来る

これは次のような事例である．
あああ“こんなかんじ”

この事例では本来ならば行末禁則文字である開きダブルクォートが行末に来ている．これは開きダブルクォートが全角文字についている為，全角の禁則処理が適用されてしまったという事例である．全角の禁則処理には半角の開きダブルクォートの禁則処理に関して記述されていない為，開きダブルクォートは分割可能と判断されてしまった．この，対策方法としては全角の開き及び閉じダブルクォートを使うという方法がある．先程の例文中のダブルクォートを全角にして表示する．
あああ“こんなかんじ”

31.2　英文が右にはみ出す

以下にはみ出る英文の例を示す．
Lopadotemachoselachogaleokranioleipsanodrimhypotrimmatosilphioparaomelitokatakechymenokichl
siraiobaphetraganopterygon.

欧文は空白のみ分割可能である為，例文のような空白が全くない場合右に文字がはみ出る．

32　jsbook クラスの左綴じ，右綴じ

jsbook クラスは製本することを前提としている為，余白が左右で異なる．両開きのページならば，左右交互に余白が変わる．openright オプションを使うことで章の始まりは必ず右ページに来る．

```
1    \documentclass[openright]{jsbook}
```

章の始まりがどちらに来てもよいという場合は openany オプションを使う．ただし，必ず左ページに来るというオプションはないため，以下のようにして左右の余白を交換する．ただし，バッドノウハウの為非推奨である．

```
1    \newlength{\tmphogehoge}
2    \setlength{\tmphogehoge}{\oddsidemargin}
3    \setlength{\oddsidemargin}{\evensidemargin}
4    \setlength{\evensidemargin}{\tmphogehoge}
```

33　Beamer で only コマンド内に vebatim なコマンドまたは環境を含む部分の注意

Beamer では表示指定子の \only を使う場合が多々ある．

```
\only<1>{1 番目に表示}
\only<2>{2 番目に表示}
\only<3>{3 番目に表示}
```

しかし，以下のように only コマンドの中に verbatim な環境やコマンドなどを入れると，エラーが起こる．

```
\only<1>{1 番目に表示}
\only<2>{
  \begin{verbatim}
    2 番目に表示
  \end{verbatim}
}
```

よって，only コマンドと同じ効果を持つ onlyenv 環境を用いる．使い方は簡単で only コマンドで囲んでいた部分を \begin{onlyenv} \end{onlyenv} に置換するだけである．

```
\only<1>{1 番目に表示}
\begin{onlyenv}{2}
  \begin{verbatim}
    2 番目に表示
  \end{verbatim}
\end{onlyenv}
```

34　newminted で定義した環境にオプションを付けたい場合

ソースコードを記載する環境として minted 環境があり，自分のオプションを盛り込んだ新しい環境を定義することが可能である．

```
\newminted{myminted}{c}{linenos}
\begin{document}
  \begin{myminted}
    #include<stdio.h>
    int main(void){
      puts("hello");
      return 0;
    }
  \end{myminted}
\end{document}
```

しかし，新しく定義した環境に加えてオプション引数を付けることはそのままではできない．オプション引数を付けたい場合は新しく定義した環境名にアスタリスクを付け，オプション引数としてではなく，第二引数として記述することでオプションを加えることができる．

```
\begin{myminted*}{fontsize=\Large}
  #include<stdio.h>
  int main(void){
    puts("hello");
    return 0;
  }
\end{myminted*}
```