基礎演習III テキスト
part 3: 文書清書システム (LaTeX) を使ってみよう

三木　邦弘

平成7年12月27日

目 次

1 はじめに

ここで紹介する文書清書システムは、ワークステーションではもっぱらワープロ代わりに使われているものです。普通のワープロとは全く異なった視点から作られています。

普通のワープロでは、画面上にあるようにプリントされます。見たままが得られるのでわかりやすいものです。逆にここで紹介するソフトは文書の構造的情報を指示して、清書はシステム任せにしますので、わかりにくいです。慣れれば使いやすい物ですが、これは学生をいじめるためにやるのだと言われると否定できない所もあります。

ワークステーションの基本ソフトであるUNIXはパソコンよりも長い歴史を持っています。画面に表示して、と言う今では何の不思議もない事が、かつては不可能もしくは非常に高価な機器を必要としていました。プリントするときに綺麗に出れば良い、かつ普通の文字が入力できれば使えると言う文書清書システムがこのような状況で生まれました。

現在は時代が変わったのでもう不用？と言えば、

- ワープロがまともに使えない人が多い。表題を用紙の中央に持ってくるのに、せっせとスペースを入れて目で確認している人が居る。
- 章や節に分かれた長い文章(卒論など)を作成しようとすると、ワープロは章の番号の付替等をしてくれないので、編集が面倒になる。
- 文章の形式を登録する機能があるので、うまく利用すれば文書の作成に専念できる。

逆に特にワープロ専用機などは、短い文章やポスターのようなものを作成するのには向いていると思います。パソコンの場合、ワープロソフトを購入すればワープロとして使えるし、ここで紹介する清書ソフトも利用できます。やはり個人用にはパソコンが一番でしょうか？

ワークステーション上で文書を清書する際に最もよく使われているTEXについてここでは説明します。実際はTEX自身を扱うのは難しいので、これを簡単に使えるようにしたLATEXを使用します。

なおこのテキストは鈴木裕信氏(hironobu@sra.co.jp)が書かれた「ひろのぶのLATEX入門 Version 0.23」を利用して作りました。このオリジナルはネットワークで公開されており、後に書籍として出版もされています。これを大幅に縮小割愛したのがこの章です。よって文章表現にやや不統一がありますし、例題等はそのまま頂いていることをあらかじめお断りしておきます。

1.1 LATEXとは

LATEXは本、レポート、記事、手紙などを書いて印刷するための仕組みです。LATEXの読み方はラテックです。 レスリー・ランポート(Leslie Lamport)が作りました。このテキストもLATEXで書かれています。文章が中身はともかくとして、美しくならんでいますね。章や節に正しく数字を振ったり、1行の文字を揃えたりするのはLATEXが行なっているのです。 **LATEXは簡単に高品質の印刷物を作る**のに便利です。

目的は同じでもワープロとはちょっと違います。ワープロは「入力すること＝清書すること」ですが、LATEXはその規則に則って書かれた原稿をプログラムにかけて初めて清書される(美しく配置される)のです。LATEXの規則は、文の論理的な構造をもとにして、全体の文字をまとめあげようとします。要するに**文章を書く時の論理的なブロック**として構成して行くのと同様な単位で処理していきます。

「LATEXやTEXはワープロみたいなWYSIWYGじゃないから難しい。」と言う指摘はあります。“**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et” (見えた通りのものを得ることが出来る)というのは裏を返せば、“what you see is all you’ve got” (結局は見えた通りのものしか得られない)ということです。ですから、センスの無い人がワープロを使うとセンスの感じられない体裁のものができます。何十ページにもおよぶ文書を作成す

る際に、最初から終わりまで文章のスタイルを一貫させることができない人がワープロを使って作成すると、内容とその表現の不一致が読者を悩ませるようなものができるでしょう。

ランポートはLaTeXを設計するにあたり次のようなことを考えました。

- 印刷の活字を組んで、どのようにレイアウトするかは著者のアイデアを読者に理解させるような形でなければならない。
- ドキュメントは読むのに楽でなくてはならない、視覚的な構造は論理的な構造を反映させていなければならない。

そして、書く時の心構えについてこう述べています。

> あなたが、文書を書く時は、視覚的にどう表現されるかではなく、論理的構造に対して配慮しなさい。

文章とは**視覚的概念ではなく論理的な概念**であるべしというわけです。さらに、文章を論理的構造に従って書いていくと、形式を変更する際や、できた文章をデータベースに入れる際に便利などの利点もあります。

LaTeXはドナルド・クヌース (Donald Knuth) の作ったTeX(テックと読む) を土台として作られているプログラムです。おいしいイチゴケーキをLaTeXとすると、土台となる軟らかいスポンジケーキがTeXというような関係にあります。TeX自体は表現力は豊かなのですが、初心者には使い難いという欠点があります。しかし、TeXはプログラミング言語としてより多くの機能を持ったシステムであり、熟練者がそれを基礎にし新しい処理系を作れるように設計されています。

LaTeXに関しては様々な図書が出版されていますが、一番頼りになるのは、LaTeXを作った本人による Leslie Lamport. LaTeX*: A Document Preparation System.* Addison-Wesley , 1986. です。幸いにしてこの訳本がアスキーより「文書処理システム LaTeX」として出版されています。

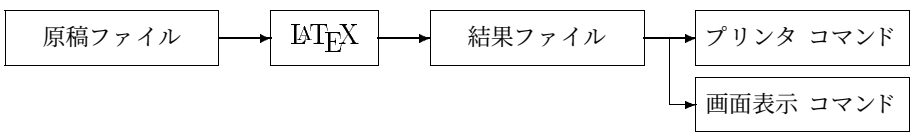
2 LaTeXの動かし方

LaTeXは現在では大型計算機からパソコンまで様々な機種で利用することができます。元来アメリカで作られたソフトウェアでしたので、当初日本語の文章には対応できませんでした。現在多くの人の努力により、日本語化されたLaTeXには次の２種類があります。

- ASCIIの日本語LaTeX
- NTTの jLaTeX

英語の部分に関してはオリジナルのLaTeXもASCIIの日本語LaTeXもNTTのjLaTeXも同じ動作をしますが、日本語の部分は完全に互換ではありません。このテキストはASCIIの日本語LaTeXを基に書いています。

処理の大まかな流れは、次のようになります。最初にLaTeXの形式で書いた原稿ファイルを作ります。LaTeXのコマンドに原稿ファイルを与えると、結果ファイルが作成されます。結果ファイルをプリンタへの出力を行なうためにコマンドに与えて実際のプリントアウトを行ないます。



2.1 コンパイル

原稿ファイルを結果ファイルに変換することをコンパイルと呼んでいます。プロンプトが出ている状態で platex ファイル名△と入力します。原稿ファイル名には**拡張子.tex がないと LaTeX に怒られます**が、コンパイルする時のファイル名は、例えばファイルが foo.tex という名前だとすると、foo というように省略しても構いませんし、foo.tex と与えても構いません。

コンパイルをすると、原稿ファイル中の、相互参照、目次、索引などの情報をまとめておくファイルが自動的に現在実行中のディレクトリに作られます。原稿ファイルの中にこれらの機能を用いた場合、LaTeX は参照できなかったラベルに出会った時と、すべての処理を終えた時に警告文を出します。よって目次などの機能を用いた場合にはコンパイルを 2 度実行する必要があります。

間違えてファイル名を指定する時がよくあります。そのような時 LaTeX は正しいファイル名を入力するようにしつこく聞いてきます。この状態の時、正しいファイル名を指定することなく抜けたい場合は**ファイル名のかわりに** null と入力します。すると、LaTeX がプロンプトを返してくるのですかさず、**\bye か CNTL–D** を入力して下さい。LaTeX から抜けることができます。

処理途中でエラーが発生した場合は x を入力すると終了します。または e を入力するとエディタを起動してエラーの発生した行へカーソルを移動してくれます。エラーが発生した場合、LaTeX はエラーメッセージと共にどうするかを聞いてきます(入力待ちになる)。なかなか判りにくいメッセージですが、良く見てから x 又は e を入力して誤りを訂正してください。

結果ファイルの名前は原稿ファイルの名前の拡張子.tex の部分が.dvi と替わったものです。原稿ファイルが foo.tex ならば**結果ファイルは** foo.dvi **です。**これは、プリンタに出力するためのコマンドを使って印刷するためのものであって、普通 DVI ファイルと呼ばれています。DVI は **D***e***V***ice* **I***ndependent format* で、「出力すべき装置とは独立した形で情報を持っている」というような意味です。

2.2 画面への出力

ワープロと異なり LaTeX では原稿と出力の形態が全く異なります。ですから本当にきれいな出力が得られるかどうかは、出力して見なければ判りません。でも、毎回プリンターで印刷したのでは資源の無駄になってしまいます。

画像を出力できる端末ですと印刷するのと同じ様な感じで表示してくれるコマンドがありますが、現在利用できる端末は文字しか表示できない端末ですので、雰囲気程度しか判りません。それでも最初のうちはできるだけ画面で出力を確認してから印刷するようにして下さい。

プロンプトが出ている状態で、jdvi2tty ファイル名△と入力すると画面に出力に近いものが表示されます。[スペース] で続きを、[b] で前の方を見る事ができ、[q] でこのコマンドを終了する事ができます。

2.3 プリンタへの出力

学園センターのプリンタに出力する時は、dviout というコマンドを使います。004 室のプリンターで出力する場合には dviprt というコマンドを使いますが、使い方は全く同じです。プリンターに出力するコマンド名は計算機の機種やプリンターの機種により様々な物があるので、他の環境ではおそらく異なるコマンド名でしょう。でもだいたい同じような機能を持っています。

dviout は DVI ファイル(先程まで結果ファイルといっていたもの)をレーザープリンタに出力します。プロンプトが出ている状態で、dviout オプション ファイル名△と入力します。

オプション[1]の部分には次のようなものを書くことができます。何も指定をしなければ、全てのページが

[1] dviout コマンドは内部的には jdvi2kps コマンドを呼び出しているので、これ以外に使える豊富なオプションに関しては jdvi2kps の man を参照してください。同じ物がそのまま使えます。

印字されます。

-f #	#ページから印字する。	-t #	#ページまで印字する。
-l	用紙に対して横方向に印字する。	-mf	手差しした用紙に印字する。

ですから全てのページを印字する場合には、dviout ファイル名△と入力し、5 ページから 8 ページを出力する場合には、dviout -f 5 -t 8 ファイル名△と入力することになります。

2.4 LaTeX 実行時に作られるファイル

LaTeX は入力や出力、自分自身が使う情報を蓄えておくファイルなどたくさんのファイルを使用します。実際に私達が扱うファイルは foo.tex か foo.dvi ぐらいですが、1 度 LaTeX を実行すると**色々な拡張子**を持ったファイルが実行時のディレクトリに作られます。

例えば、拡張子 aux のついたファイルには相互参照表、索引、図目次、表目次などを作る時に必要な情報を書き出します。また、拡張子 log のついたファイルは LaTeX が実行中に画面に出力していた内容などを収めています。

これ以外には次のようなファイルが実行時の条件により作成されます。\tableofcontents のコマンドがあった時には、目次を作るために拡張子 toc のついたファイルが作られます。\listoffigures のコマンドがあった時には、図目次を作るために拡張子 lof のついたファイルが作られます。\listoftables のコマンドがあった時には、表目次を作るために拡張子 lot のついたファイルが作られます。\makeglossary というコマンドが使われた時には、\glossary というコマンドによって作られた情報が拡張子 glo のついたファイルに入り、用語索引ができます。\makeindex というコマンドが使われた時には、\index というコマンドによって作られた情報が拡張子 idx のついたファイルに入り、索引ができます。

演習問題

次のような文章を入力し、コンパイルし、プリンターに出力してみよ。なお、「ひろのぶ」の部分は自分の学籍番号と名前に直すこと。

```
\documentstyle[a4j]{jarticle}
\title{SF 桃太郎}
\author{ひろのぶ}
\begin{document}
\maketitle

昔むかし、おじいさんと、おばあさんが住んでいました。

おじいさんは、クレーターヘリチューム結晶を掘りにいきました。
おばあさんは、氷河に洗濯をするための水源となる氷をとりにいきました。

すると、空から氷河に救助用小型カプセルがおりてきました。
表面は焼けただれ、一部は解けてさえいました。

おばあさんはそれを家に持ち帰り、おじいさんと二人で分解しました。
すると、カプセルのなかから生まれてすぐの男の子が現れました。
\end{document}
```

3 基本的な書き方

ここでは LaTeX の書き方について説明を行ないます。良く使う命令の書き方と、実際の出力結果のサンプルを載せます。細かい説明については省いています。

3.1 使用する文字

入力に使用できる文字は全角文字、半角の英数字、そして、次の16の特殊文字(puncturation)です。

. : ; , ? ! ‘ ’ ( ) [ ] - / * @

そして、次の10個の特殊文字は、LaTeXの命令に使用されます。ですからそのまま使用するとどんでもない意味に解釈される事があります。

$ % & _{ } ~ ^ \

\以外は\#、\$とすることによって文字として表現できます。ただし、バックスラッシュ\は次のようにキャラクタシンボルを表現する方法で行なうことが必要です。

$\backslash$ ⟹ \

LaTeXはテキスト中に空白があっても自動的につめてしまいます。意図的に空白を挿入する時は、\␣する必要があります。改行の場合は\\を使います。マイナス記号(-)は特別な働きをします。連続して書くことによってハイフンなどの線(- – —)を引くことができます。例えば—このような時に便利です。注意しなければならないのは、マイナス記号が1つのときは、マイナス記号が2つ、3つ連続している時とはちょっと違うことです。次の例を良く見て下さい。

- -- --- ---- ----- ⟹ - – — —- —

3.2 特殊文字の表記

フランス語やドイツ語等の文字には、アクセント記号が付いたものがあります。以下はそれら文字の表し方です。例えば、évêque は\’{e}v\^{e}que と記述します。

ò \‘{o}	õ \~{o}	ǒ \v{o}	o̧ \c{o}
ó \’{o}	ō \={o}	ő \H{o}	ọ \d{o}
ô \^{o}	ȯ \.{o}	o͡o \t{oo}	o̲ \b{o}
ö \"{o}	ŏ \u{o}	å \aa	Å \AA
æ \ae	Æ \AE	ł \l	Ł \L
ø \o	Ø \O	œ \oe	Œ \OE
ß \ss	¿ ?‘	¡ !‘	

3.3 コマンドと環境

LaTeXにはコマンド(command)と環境(environment)の考え方があります。コマンドは局所的、環境は広域的に影響を及ぼします。

コマンドは~ や \で始まります。コマンドは{}や[ ]で囲まれた引数を取ることができます。たとえば、脚注を書くコマンドは\footnote{}であり\footnote{脚注です。}と書けば、実際には現在のページの下[2]に書かれます。

\beginや\endもそのような意味ではコマンドです。注意しなければならないのは、コマンドを識別子としてあつかう必要があるので、\LaTeXなどは\LaTeX␣というようにしなければなりません。

環境とは\begin{*name*}と\end{*name*}で囲まれた範囲が、*name*と言う規則にすべて従うという感じです。

[2]脚注です。

3.4 原稿の書き方

原稿は必ず \documentstyle[*option*]{*style*} で始めなければなりません。そして文章の本体は \begin{document} と、\end{document} で囲まれた範囲内に書かなければいけません。

原稿のスタイル (*style*) の標準的なものには次のようなものがあります[3]

jbook これは、本の形式 (スタイル) です。奇数ページの上部 (ヘッダ) では節 (セクション) の番号とセクション名が左側、ページ番号が右側に出ます。偶数ページヘッダでは右側に章 (チャプタ) 名、ページ番号が左側に出ます。章の最初のページが奇数になるようにページが送られます。章立ては必ず \chapter から始めなければなりません。もし、\section レベルから始めた場合、エラーとなります。

book これは英文の本のスタイルです。

jreport これは報告書の形式です。ページ番号は用紙の下部の中心にくるような形になります。章立ては必ず \chapter から始めなければなりません。もし、\section レベルから始めた場合、エラーとなります。

report これは英文のレポートのスタイルです。

jarticle これは最もよく使われる論文の形式です。このテキストもこのスタイルを使っています。\section のレベルから書くことができます。

article これは英文の論文のスタイルです。

オプション (option) はスタイルの補助的な役割をします。基本となるフォントサイズの変更や見開きに変更する事ができます。

11pt 基本フォントサイズを 11 ポイントにします。標準のフォントサイズは 10 ポイントです。

12pt 基本フォントサイズを 12 ポイントにします。

twoside 見開きを行なうためのスタイルです。*book*、*jbook* スタイルは標準でこうなっています。

twocolumn 2 段組のページにします。

titlepage *article*、*jarticle* にのみ有効です。このオプションを指定すると、\maketitle や abstract 環境を使った結果が独立した紙に出されます。(要するに表紙ができる。)

LaTeX は指定したスタイルのスタイルファイルと呼ばれているもの (*style*.sty) を最初に読み込みます。

3.5 タイトルページ

タイトルページは \maketitle があらわれた時点でタイトルページが作成されます。

```
\title{HOW TO USE \LaTeX}
\author{Hironobu Suzuki}
\date{4 Oct 1988}
.....
\maketitle
```

⟹

HOW TO USE LaTeX
Hironobu Suzuki
4 Oct 1988

[3] スタイルの名前で先頭に **j** がついているものは日本語 LaTeX 専用のものです。オリジナルの英語の LaTeX で使用すると、エラーとなりますので注意して下さい。

3.6 文章構成

文章の構成で、章や節を表すものとして、次のようなものが用意されています。

```
\part           \section            \paragraph
\chapter        \subsection         \subparagraph
                \subsubsection
```

各々は1つの引数を取ります。通常の jarticle スタイルでは、節 (section) の中が項 (subsection) に分かれ、さらにその中が部分項? (subsubsection) に分かれます。jbook や jreport は章 (chapter) の中が節に分かれ、以下 jarticle と同様に細かく分かれて行きます。この際引数として与えるのはそれぞれの見出しで、番号は LaTeX が自動的に付けてくれます。文章を考えながら作成する際には、章や節の順番が入れ替わる事はよく生じます。ワープロであればその度に章や節の番号を自分でつけ直す必要がありますが、その作業を行う必要がないために文章を考える事に専念できます。

よってこの項の原稿には、`\subsection{文章構成}`と書いてあるだけで、それにどんな番号が付くかはできてからのお楽しみであります。

3.7 空白

半角の空白は複数あっても、1つの空白と見なされます。逆に全角の空白は目に見えませんが、行の前後に残っていると歯抜けの段落ができたりするので注意が必要です。

意図的に空白を挿入したい場合は`\␣`とします。これは`\LaTeX on UNIX`のような書き方をしたとき LaTeXon UNIX というような結果になってしまいます。このような結果を避けたい時は`\LaTeX\ on␣UNIX`とすれば LaTeX on UNIX となります。

3.8 行、段落

原稿の文字列は自動的に詰められ改行されたりします。空行は段落の切れ目を意味しますが、複数の空行は1行分の空行と解釈されます。段落の先頭は1字下げがなされるので、原稿では字下げは不要です。

```
左につめて書きます。ただし、前の行は空白行です。

連続した行です。
連続した行です。
連続した行です。
連続した行です。
連続した行です。
連続した行です。
連続した行です。
連続した行です。
```

⟹

左につめて書きます。ただし、前の行は空白行です。

連続した行です。連続した行です。連続した行です。連続した行です。連続した行です。連続した行です。連続した行です。連続した行です。連続した行です。

3.9 改行

改行を行ないたい場合は`\\`か、`\newline` を使用します。その働きは正確に言えば段落の切れ目を示す事ですので、後で出てくる表での使用以外の本文では、これの代わりに空行を用いた方が良いでしょう。

逆に文字列や単語を途中で自動的に切り離されたくないとき、`\mbox` を使います。例えば、*Segmentation fault* の *Segmentation* と *fault* を離したくない場合は`\mbox{`*Segmentation fault*`}`とします。実際はコンパイルして、印刷して見て変な場所だけにこのテクニックは使ったほうが良いでしょう。

3.10　文字の大きさ

文字の大きさは\Largeのようなコマンドを使って変更します。直接ポイント数を与えて文字の大きさを変えるような野蛮なことはしません。largeやsmallのコマンドによって文字のポイント数が変わりますが、原稿の基準フォントのポイントサイズに合わせて相対的にlargeであり、smallであります。このように総体的に指定する事により、後になって基本となるサイズを変更した場合にも、本文の指定を変更する必要がなくなります。

文字の大きさの有効範囲は同一ブロック内です。ブロックとは、環境の中や{と}で囲んだ範囲の事です。例えば、{\Large abc}dと行なうとabcが大きくなりdは元のサイズのままです。実際に行なってみるとabcdとなります。もちろん、入れ子にでき、{\Large a {\LARGE b {\small c}}}はa b cというようになります。

ワープロの初心者が作成した文書などで、頻繁に文字の大きさを変える例があります。必要以上の文字のサイズ変更は見やすさよりも見にくさを助長するだけなので避けるようにしましょう。

文字 \Huge
文字 \Large
文字 \huge
文字 \large
文字 \LARGE
文字 \normalsize
文字 \small
文字 \footnotesize
文字 \scriptsize
文字 \tiny

3.11　文字のスタイル

英数字の文字のスタイルは全部で5つありますが、全角に関しては太字のみです。そのため、スタイルの指定を行なっているブロック内に全角文字があっても太字以外にはなりません。

影響を与える範囲などは文字のサイズの時と同じです。\emを用いて影響の及ぶ範囲の例を示します。

ここでemphasizeの指定をスタイルと同等に扱っていますが、LaTeXの作者であるランポートの本によれば、下線とイタリック文字というのは、視覚的な概念であるのに対し、強調(emphasize)は論理的な概念であると述べられています。

使える文字スタイルは次のようになります。

```
{\bf Bold type style. 全角は太字です。}
{\sf Sans serif type style. 全角は同じです。
}
{\sl Slanted type style. 全角は同じです。}
{\tt Typewriter type style. 全角は同じです。
}
{\sc Small caps type style. 全角は同じです。
}
{\em A emphasized line. 全角は太字です。}
\underline{underlined text} \underline{架
線付き文字です。}
```

⟹

Bold type style. 全角は太字です。 Sans serif type style. 全角は同じです。*Slanted type style.* 全角は同じです。Typewriter type style. 全角は同じです。SMALL CAPS TYPE STYLE. 全角は同じです。*A emphasized line.* **全角は太字です。** underlined text 架線付き文字です。

最後の下線の例はついでのオマケです。これはコマンドなのでちょっと形式が異なる所にご注意下さい。

3.12　行の離し方

段落の前後を離したいときや表とその説明の部分を空けたいときは、次のようなコマンド、\smallskip、\medskip、\bigskipを使用します。実際の間隔はページのレイアウトに左右されますので、同じコマンド

でもいつも同じ間隔が空くとは限りません。

行間や段落間の間隔を調整する直接指定する方法として\vspaceなどのコマンドがあります。既に書式が決まっているものに無理矢理合わせるような場合以外は使わないほうが無難です。

```
前の行は空白行です。

\smallskip
前の行は空白行です。

\medskip
前の行は空白行です。

\bigskip
前の行は空白行です。
```

⟹

前の行は空白行です。

前の行は空白行です。

前の行は空白行です。

前の行は空白行です。

\vspaceの使い方です。意味は垂直方向への空白です。inとあるのは長さの単位がインチであることを示しています。長さの単位にはこのほかに、cm、mm等も使えます。

```
This \vspace{.5in}line exist for
testing vspace command. Look carefuly
between first line to second line.

\vspace{7mm}

Interpreter: One who enables two persons
of different languages to understand each
other by repeating to each what it would
have been to the interpreter's advantage
for the other to have said.
```

⟹

This line exist for testing vspace command. Look

carefuly between first line to second line.

Interpreter: One who enables two persons of different languages to understand each other by repeating to each what it would have been to the interpreter's advantage for the other to have said.

次は\hspaceの使い方です。意味は横方向の空白です。前後の文字との距離が微妙に変化します。

```
1inch\hspace{1in}space.
1inch \hspace{1in}space.
1inch \hspace{1in} space.
This command can accept minus
 paramater like \hspace{-1in}/////////
```

⟹

1inch space. 1inch space. 1inch space. This command can accept minus paramater like /////////

3.13 中ぞろえ

centerは文章を中央に持ってくる時に使われる環境です。

```
\begin{center}
ここに配置しているものは、中央に寄せられます。

中央
\end{center}
```

⟹

ここに配置しているものは、中央に寄せられます。

中央

3.14 右ぞろえ

flushrightは文章を右にそろえる時に使われる環境です。

```
\begin{flushright}
ここに配置しているものは、右に寄せられます。

右寄り
\end{flushright}
```

⇒

ここに配置しているものは、右に寄せられます。
右寄り

3.15 引用

quotation や quote は文章をまとめて引き下げることを行ないます。この環境で囲んだ部分は、前の行よりも字下げし、かつ行の長さも短くなります。前者と後者との差は段落の最初を字下げするかしないかです。

```
この例題は quote がどのようにインデントするかを示すものです。
\begin{quotation}
ここから、一段インデントしています。
この環境にいる間はずーっとインデントされています。
\begin{quote}
ここからまた、インデントされています。
通常環境は入れ子にしても大丈夫です。ただし、後に述べる verbatim だけ
は入れ子にできません。
\end{quote}
\end{quotation}
```

⇒

この例題は quote がどのようにインデントするかを示すものです。

> ここから、一段インデントしています。この環境にいる間はずーっとインデントされています。
>
> > ここからまた、インデントされています。通常環境は入れ子にしても大丈夫です。ただし、後に述べる verbatim だけは入れ子にできません。

3.16 箇条書

箇条書の書式として LaTeX では 3 種類の方法があります。先頭に• をつけるもの、先頭に番号を振るものと、、先頭にラベルを指定できるものです。それぞれ、itemize、enumerate、description です。

```
\begin{itemize}
\item 各々の行はあたまにポチを付けられてならんでいるでしょ。
\item
書き出しは直接後ろからでなくても大丈夫です。
\begin{enumerate}
\item ここの行の頭には数字がついているはずです。

\item ほらね、2 行目でしょ。
\end{enumerate}

もちろん離れていてもきちんと対処してくれます。


\item 空白行は何の影響も与えません。
\begin{description}
\item [ラベル]　ここの行の頭にはラベルがついているはずです。
\item [Label]　ラベルは bold 体になっています。

\end{description}
\end{itemize}
```

⇒

- 各々の行はあたまにポチを付けられてならんでいるでしょ。
- 書き出しは直接後ろからでなくても大丈夫です。

  1. ここの行の頭には数字がついているはずです。
  2. ほらね、2 行目でしょ。

  もちろん離れていてもきちんと対処してくれます。
- 空白行は何の影響も与えません。

ラベル　ここの行の頭にはラベルがついているはずです。

Label　ラベルは bold 体になっています。

3.17 そのままの形

verbatimはテキストを入力したそのままの形で表すための環境です。例えば、%\}{はそのまま打ってもLaTeXでは正しく表示してくれません。Cのプログラムのリストをそのまま引用するような場合、この環境を使用します。

⟹

\begin{verbatim} 次のキャラクターは通常、特殊文字として扱われるので表示できない。バックスラッシュを使うかverbatimを使うかどちらかである。
\ \ \ \ ??? ~~~ $$$$ ###
\end{verbatim}

次のキャラクターは通常、特殊文字として扱われるので表示できない。
バックスラッシュを使うかverbatimを使うかどちらかである。
\ \ \ \ ??? ~~~ $$$$ ###

行中にこれらの特殊文字を組み込みたいときは、\verbを使用します。\verb+??##$$+とすると、??##$$と出力されます。

3.18 要約

タイトルページに要約に付ける時はこの環境を使います。\maketitleコマンド より前にこれを使う必要があります。

⟹

```
\begin{abstract}
This article is for the beginner.
This abstract is printed onto top page.

\end{abstract}
```

Abstract

This article is for the beginner. This abstract is printed onto top page.

3.19 表

LaTeXでは、表のようなものを作るのに、tabbing、tabular、arrayという環境が利用できます。それぞれ想定された用途があり、特徴がありますがここではtabularのみ説明をします。

tabularの引数の中での命令の意味は、lは左詰め、rは右詰め、cは中央に各項目をそろえます。p{wd}は列の幅をwdに指定(たとえばp{1cm})することにより、固定することができます。

| をつけると垂直方向の線が書けます。\\の後の\hlineは水平方向の線を引きます。\cline{i – j} はi列からj列までの水平方向の線を引きます。

⟹

```
\begin{tabular}{||l|c|r||}
\hline
sra & sra-b1-net & 12098 packets \\
      \cline{2-3}
    & sra-2-net  & 290873 packets \\
      \hline
ntt & sun-loop-back  & 2839  packets \\
      \cline{1-1} \cline{3-3}
etl &  & 287 packets \\
\hline
titech & titech-a1-net & 28329 packets \\
\hline
\end{tabular}
```

sra	sra-b1-net	12098 packets
	sra-2-net	290873 packets
ntt	sun-loop-back	2839 packets
etl		287 packets
titech	titech-a1-net	28329 packets

一つ一つの項目欄は\multicolumn{n}{pos}{item}によって調整することができます。nは使用する列数、posは項目の水平方向の位置、itemは項目の文字列です。引数であるposは現在すでに設定されているものを無効とし新たに設定します。この時有効となる引数posはl r c | pからなるものでなくてはいけません。

```
\begin{tabular}{|l|l|r|} \hline \hline
{\em team } &
  \multicolumn{2}{c|}{{\em driver}} \\ \hline
マクラーレン & セナ &プロスト \\
ロータス & ピケ &  中島\\ \hline \hline
\end{tabular}
```

team	*driver*	
マクラーレン	セナ	プロスト
ロータス	ピケ	中島

3.20 数式

数学モード (math mode)は数式を表現するために使用するモードです。TEXは元もと数学の本を美しく作るために生まれたために、数式の表現は得意です。しかし、一般的には数式を必要とすることは少ないのでここでは簡単な例を示すだけにします。

文中とは独立して数式を示したいときには、\begin{math} ...\end{math} を使います。文中に数式を入れる場合には$...$を使います。例えば、$ \int \sin(2x+3)\cos(x)^{2} dx $と記述すると、$\int \sin(2x+3)\cos(x)^2 dx$のように出力されます。

```
\begin{math}
 \int  \frac{x \cos(x) }
       { \sin(x)^{2}} dx
\end{math}
```

$$\int \frac{x \cos(x)}{\sin(x)^2} dx$$

数式中での英字はxyというように斜体になります。一方関数名などは例えば、logは通常の字体でなければなりません。そこで\log を使用します。例えば、$\log x^{y} = y \log x $とすれば$\log x^y = y \log x$となります。

$\cos(x)$	\cos(x)	$\exp(x)$	\exp(x)
$gcd(x,y)$	\gcd(x,y)	$\log(x)$	\log(x)
$\max(x,y)$	\max(x,y)	$\min(x,y)$	\min(x,y)
$\sin(x)$	\sin(x)	$tan(x)$	\tan(x)
A^b	A^{b}	A_b	A_{b}
A_b^c	A_{b}^{c}	$\frac{x}{y}$	\(\frac{x}{y} \)
$\sqrt{A}$	\(\sqrt{A}\)	$\sqrt[b]{A}$	\(\sqrt[b]{A}\)

その他に数学の記号として次のようなものも利用できます。

$\pm$ \pm	$\mp$ \mp	$\times$ \times	$\div$ \div
$*$ \ast	$\star$ \star	$\leq$ \leq	$\ll$ \ll
$\geq$ \geq	$\gg$ \gg	$\sim$ \sim	$\approx$ \approx
$\neq$ \neq			

3.21 様々な記号

文中では以下の様な記号を利用する事ができます。なお、これらも数学の記号の扱いなので、通常の文中では$\leftarrow$のように$マークで囲む必要があります。

$\leftarrow$ \leftarrow	$\Leftarrow$ \Leftarrow	$\rightarrow$ \rightarrow
$\Rightarrow$ \Rightarrow	$\leftrightarrow$ \leftrightarrow	$\Leftrightarrow$ \Leftrightarrow
$\rightharpoondown$ \rightharpoondown	$\leadsto$ \leadsto	$\uparrow$ \uparrow
$\Uparrow$ \Uparrow	$\downarrow$ \downarrow	$\Downarrow$ \Downarrow
$\updownarrow$ \updownarrow	$\Updownarrow$ \Updownarrow	$\nearrow$ \nearrow
$\searrow$ \searrow	$\swarrow$ \swarrow	$\nwarrow$ \nwarrow
$\flat$ \flat	$\natural$ \natural	$\sharp$ \sharp
$\backslash$ \backslash	∞ \infty	$\Box$ \Box
$\Diamond$ \Diamond	$\triangle$ \triangle	$\clubsuit$ \clubsuit
$\diamondsuit$ \diamondsuit	$\heartsuit$ \heartsuit	$\spadesuit$ \spadesuit
α \alpha	β \beta	γ \gamma
† \dag	§ \S	© \copyright
‡ \ddag	¶ \P	$\pounds$ \pounds

ちょっと特殊な記号？として\LaTeX、\TeX 、\ldots や\today があります。これらはそれぞれ、LaTeX、TeX、...や平成 16 年 1 月 22 日になります。

演習問題

1. 「SF 桃太郎」を章分けして続きを追加せよ。前の課題の部分を「はじまり」として、「桃太郎の成長」、「桃太郎の旅立ち」と言う章を追加せよ。追加した章の内容は適当に作成せよ。全体として 1 ページ程度にすること。