

L^AT_EX 入門

目次

1	T _E X とは	1
2	はじめの一步	2
3	L ^A T _E X 文書の書き方	2
4	数式を書く	4
5	L ^A T _E X 文書の構造	6
6	いろいろな環境	7
7	図の挿入	9
8	参考文献とその参照	10

1 T_EX とは

T_EX(日本では「てふ」と発音する人が多い)は、UNIX 上で文書を綺麗に印刷するときに最もよく用いられるものである。D. E. Knuth という計算機科学で有名な先生が、自分の本を出版するときの組版の品質の悪さに憤慨して、一流の活字工の技術を後世に残したい一心で T_EX を作成した。フリーソフトウェアである。

T_EX は、MS WORD などと違い、文字装飾等の情報を文章中に埋め込む形をとる。例えば、

Input:

ここは{\small 小さな文字}、あそこは{\large 大きな文字}

のように書くと、

Output:

ここは小さな文字、あそこは大きな文字

のようになる。一種のプログラム言語になっていて、コマンドを定義して自分自身を拡張することも出来る。

コマンドを自分で定義できることを利用して、L. Lamport という人が、文書の構造を表現することが出来るように T_EX を拡張したものが、L^AT_EX(日本では「らてふ」と発音する人が多い)である。生の T_EX で文書を書くことは少なく、普通 L^AT_EX を用いる。

また、T_EX には長い歴史があり、日本語化した pT_EX(pL^AT_EX)、多国語化した upT_EX(upL^AT_EX)、lua と呼ばれるスクリプト言語を組み込んだ LuaT_EX(LuaL^AT_EX) など、多くの拡張が作られている。ここでは、最新かつ今後主流になると見られている、LuaL^AT_EX に基づいて説明する。

2 はじめの一步

(La)TeX における文章作成の簡単な作業の流れを以下に示しておく。

```
test.tex
\documentclass[a4j]{ltjsarticle}
\begin{document}
  LaTeX をはじめよう
\end{document}
```

1.Editor まず適当な editor を用いて、拡張子に“tex”がついた tex ファイルを作る。

```
vi test.tex
```

2.Compile 次に、tex ファイルを pdf ファイルに変換する。(また、この作業により pdf ファイル以外に aux ファイル, log ファイル等が生成される)

```
lualatex test.tex
```

ここで、後で述べる相互参照機能を用いている場合は、2度実行する。目次を使う場合にはさらにもう一度。

ただし、具体的なコマンド名等は環境により異なる。

以上の作業において、1 の LaTeX ファイルの作成はプログラムの作成にあたり、2,3 の pdf ファイルへの変換がコンパイルに相当する。

コンパイル時にエラーメッセージの後“?”が出て止まってしまった場合は、

```
x
```

とすれば TeX を終了できる。それ以外の状態で止まってしまった場合は、Ctrl-D で終了できる。

3 LaTeX 文書の書き方

以下の特殊記号はそのままタイプしても TeX では表示されない。それどころか、コンパイルさえ通らないことも多いので、とりあえず表示する方法を表にしておく。

特殊記号	コマンド	意味
\	<code>\backslash</code>	コマンドの始め
{,}	<code>\{, \}</code>	グループの始めと終り
\$	<code>\\$</code>	数式モードの始めと終り
&	<code>\&</code>	タブ
#	<code>\#</code>	マクロの引数
^	<code>\^{} </code>	上付添字作成
_	<code>_{} </code>	下付添字作成
%	<code>\%</code>	コメント
~	<code>\~{} </code>	単語間空白

3.1 注釈

LaTeX 文書の中では、% から行末では無視されるので、% を用いて注釈・覚書などを書くことが出来る。

Input:

```
LaTeX を % はじめよう
```

Output:

```
LaTeX を
```

Input:

```
% 1966 年 12 月 22 日加筆
今日は誰かの誕生日！
```

Output:
今日は誰かの誕生日！

3.2 改行

L^AT_EX 文書中の 1 回のみの改行は次のように無視される。

Input:
LaTeX を
はじめよう

Output:
LaTeX をはじめよう

しかし、2 回続けての改行 (空白の行の挿入) では新しい段落の開始を意味し、次のように改行されて字下げが行われる。

Input:
LaTeX を

はじめよう

Output:
LaTeX を
はじめよう

また、明示的に改行するためには `\\` を改行場所に入れる。

Input:
LaTeX を\\
はじめよう

Output:
LaTeX を
はじめよう

3.3 空白の制御

- 小さな空白は`\`,
逆方向には`\!`
- 大きな空白
横方向
`\hspace{長さ}`あるいは`\hspace*{長さ}`
縦方向
`\vspace{長さ}`あるいは`\vspace*{長さ}`
この長さには、3mm, 1cm, -8mm などが入る。

3.4 フォント

L^AT_EX で標準的に用いられる英字フォントとして以下のものが用意されている。

<code>\rm roman</code>	roman
<code>\bf bold</code>	bold
<code>\it italic</code>	<i>italic</i>
<code>\sl slant</code>	<i>slant</i>
<code>\tt typewriter</code>	typewriter
<code>\sc Small Caps</code>	SMALL CAPS
<code>\sf sans serif</code>	sans serif

日本語フォントの場合

<code>\mc 明朝体</code>	明朝体
<code>\gt ゴシック体</code>	ゴシック体
<code>\bf ゴシック体</code>	ゴシック体

文字の大きさ

<code>\tiny</code>	size
<code>\scriptsize</code>	size
<code>\footnotesize</code>	size
<code>\small</code>	size
<code>\normalsize</code>	size
<code>\large</code>	size
<code>\Large</code>	size
<code>\LARGE</code>	size
<code>\huge</code>	size
<code>\Huge</code>	size

フォントを部分的に換えたい場合は{.....}で
換えたい箇所を囲む。

Input: `{\bf La} {\Large\tt TeX}`を
`{\gt はじ}{\Large めよう}`

Output: **LaTeX** をはじめよう

その他の装飾として

Input: `\underline{下線がひかれています}`
`Schr\"odinger 方程式`

Output: 下線がひかれています
Schrödinger 方程式

などの文字飾りもある。

4 数式を書く

4.1 短い数式

1 行以内の短い数式を表示する方法として、
次のものがある。

1. インライン数式

数式部分を\$で囲む。

Input: 線形方程式 $a x + b y = c$ を解く

Output: 線形方程式 $ax + by = c$ を解く
線形方程式 $a x + b y = c$ を解く

2. ディスプレイ数式

数式番号なし 数式部分を $[,]$ で囲む。

Input: 線形方程式 $[ax+by=c]$
を解く

Output: 線形方程式
$$ax + by = c$$

を解く

数式番号付き equation 環境を用いる

Input: 線形方程式
`\begin{equation}`
 $a x + b y = c$
`\end{equation}`
を解く

Output: 線形方程式
$$ax + by = c \quad (1)$$

を解く

4.2 様々な数学記号

添字 上つき添字は`^`, 下つき添字は`_`を用いて表示する.

Input:

```
% 上つき添字
\[ a^2, b^{2n}, x^{y^z} \]
% 下つき添字
\[ a_2, b_{2n}, x_{y_z} \]
```

Output:

$$a^2, b^{2n}, x^{y^z}$$

$$a_2, b_{2n}, x_{y_z}$$

上つき添字と下つき添字を一緒に用いることも可能. $a^{x_{3m}}$ もしくは a_{3m}^x で a_{3m}^x と表される.

分数 命令`\frac{分子部分}{分母部分}`を用いる

Input:

```
\[ \frac{F}{G} \]
```

Output:

$$\frac{F}{G}$$

ルート 平方根は`\sqrt{.....}`,
 n の冪乗根は`\sqrt[n]{.....}`

Input:

```
\[ \sqrt{(x-x_0)^2+(y-y_0)^2} \]
\[ \sqrt[n]{(x-x_0)^2+(y-y_0)^2} \]
```

Output:

$$\sqrt{(x-x_0)^2+(y-y_0)^2}$$

$$\sqrt[n]{(x-x_0)^2+(y-y_0)^2}$$

積分・和・積記号 積分は`\int`, 和は`\sum`, 積は`\prod` で表し, 上限を`^`, 下限を`_`で指定する事ができる.

Input:

```
\[ \int_0^1 x^2 dx
+ \sum_{i=1}^N a_i
+ \prod_{i=1}^{\infty} \frac{1}{i} \]
```

Output:

$$\int_0^1 x^2 dx + \sum_{i=1}^N a_i + \prod_{i=1}^{\infty} \frac{1}{i}$$

ギリシャ文字等 数式モードの中では,
`\alpha`, `\beta`, `\gamma` などにより,
 α, β, γ などが表示される.

ベクトル記号やハット記号などの装飾を
変数に施す場合, `\vec x`, `\hat y` と表
す事により, $\vec{x}, \hat{y}$ と表示される.

4.3 複数行からなる数式

複数行からなる数式は, `eqnarray` 環境を用いる.
`eqnarray` 環境では2つの`&`マークの箇所で
上下の位置揃えがされる.

Input:

```
\begin{eqnarray}
x+y=2 \\
x+3y=5
\end{eqnarray}
\begin{eqnarray}
x+y=&2 \\
x+3y=&5
\end{eqnarray}
```

Output:

$$\begin{array}{rcl} x+y & = & 2 \\ y & = & 2-x \end{array} \quad \begin{array}{l} (2) \\ (3) \end{array}$$
$$\begin{array}{rcl} x+y & = & 2 \\ y & = & 2-x \end{array} \quad \begin{array}{l} (4) \\ (5) \end{array}$$

eqnarray 環境中で特定の行で数式番号を付けたくない場合は、`\nonumber` をその行の中に挿入する。全ての行に数式番号をつけない場合は、`eqnarray*`環境を用いる。

また、その他の環境についてもいえるが、最後の行に`\\`を入れると余計な空白が入るので最後の行に改行命令は付けない。

5 L^AT_EX 文書の構造

L^AT_EX 文書の大まかな構造は次の通り。

```
\documentclass[option]{文書スタイル}

    プリアンブル (宣言・設定など)

\begin{document}
    本文
\end{document}
```

L^AT_EX 文書の先頭の

- 文書スタイル
文書の内容に応じて

<code>ltjsbook</code>	(本の形式)
<code>ltjsarticle</code>	(論文の形式)
<code>ltjsreport</code>	(レポート形式)

下では `ltjsarticle` としている。

- option
文書スタイルのオプションをいれる。例えば、ここを `12pt` とすると、本文を 12 ポイントで印字する。`11pt` の場合は 11 ポイントにする。(ちなみに、何も書かなければ 10 ポイントで印字する) また、`a4j` により、A4 版のレイアウトを指定することが出来る。複数指定する場合は「,」で区切る。

一般に、L^AT_EX 文書の本文は章 (chapter)・節 (section) などに構造化されており、それら構造を明示するための命令 (`\section{title}`等) が用いられている。

- `\section` 命令の仲間として
`\chapter{title}`
`\section{title}`
`\subsection{title}`
`\subsubsection{title}`
`\paragraph{title}`
`\subparagraph{title}`
などがある。
- プリアンブル部に

```
\title{題名}
\author{名前}
\date{日付}
```

本文中の先頭に

```
\maketitle
```

を書くと、タイトル部が付く。

- 目次を入りたい箇所に`\tableofcontents`を挿入する。

Input:

```
\documentclass[12pt,a4j]{ltjsarticle}
\title{Title}
\author{Author}
\date{Date}
\begin{document}
\maketitle
\section{はじめに}
LaTeXをはじめよう
\section{まんなか}
LaTeXをならった
\subsection{小さな節 1}
頑張る
\subsection{小さな節 2}
さらに頑張る
\section{おわりに}
LaTeXがつかえるぞ
\end{document}
```

Input:

```
\section{Section1}
\label{sec1}
\begin{equation}
\frac{d}{dt} f(t) = C
\label{eq1}
\end{equation}
(\ref{eq1}) 式は、微分方程式です。
この章は第\ref{sec1}章です。
```

Output:

Section1

$$\frac{d}{dt} f(t) = C \quad (6)$$

(6) 式は、微分方程式です。この章は第 1 章です。

6 いろいろな環境

```
\begin{環境名}
  本文
\end{環境名}
```

5.1 相互参照機能

`\label{ラベル名}`と`\ref{参照するラベル名}`を用いて数式番号や章番号の相互参照することができる。この機能を使う場合は $\text{\LaTeX}$ 文書のコンパイルを2回する必要がある。これは、1回目のコンパイルでラベル名と番号の対応を`aux`ファイルに出力し、2回目のコンパイルで`aux`ファイルからその対応を読みとり、`pdf`ファイルに对应結果を反映させる、といった動作をしている事から来る約束事である。

6.1 中よせ, 右よせ, 左よせ

`center`, `flushright`, `flushleft`

Input:

```
\begin{center}
LaTeX をはじめよう
\end{center}
\begin{flushleft}
LaTeX をはじめよう
\end{flushleft}
\begin{flushright}
LaTeX をはじめよう
\end{flushright}
```

Output:

- 身長
- 体重

1. はじめ
2. おわり

はじめ はじめ
おわり おわり

Output:

LaTeX をはじめよう

LaTeX をはじめよう

LaTeX をはじめよう

6.2 箇条書き

itemize, enumerate, description 環境を用い、
項目の先頭には\itemを挿入して下さい

Input:

```
\begin{itemize}
\item 身長
\item 体重
\end{itemize}
\begin{enumerate}
\item はじめ
\item おわり
\end{enumerate}
\begin{description}
\item[はじめ] はじめ
\item[おわり] おわり
\end{description}
```

Output:

a	b	c
d	e	f

[...] で表の表示位置を優先順位順に指定し、
{...}で表のフォーマットを指定する。

表示位置オプション			
h	here	p	page
b	bottom	t	top

フォーマットオプション			
l	左よせ	r	右よせ
c	中よせ		縦線を引く

表中では,&で区切り指定をし,\hlineで横線を引く。

また、横の複数の項目を合わせて使いたい場合、

3つの引数をもつ次の命令を用いる。

`\multicolumn{結ぶ項目数}{format}{内容}`

例えば、前の表は

```
\begin{tabular}[htbp]{|cc|cc|}
\hline
\multicolumn{4}{|c|}{表示位置オプション} \\
\hline
h & here & p & page \\
b & bottom & t & top \\
\hline
\end{tabular}
\\
\smallskip
\begin{tabular}[htbp]{|cc|cc|}
\hline
\multicolumn{4}{|c|}{フォーマットオプション} \\
\hline
l & 左よせ & r & 右よせ \\
c & 中よせ & \verb#| & 縦線を引く \\
\hline
\end{tabular}
```

となる。tabular は一つの大きな文字と解釈され、そのままでは文中に埋め込まれる。独立に表示したい場合は center 環境などを使えばよい。

6.4 ソースをそのまま出力

プログラムなどを表示する場合、ソースをそのまま出力するための環境がある。

verbatim, verbatim*

Input:

```
\begin{verbatim}
#include<stdio.h>

int main(void)
{
    int i,j;
}

\end{verbatim}
```

Output:

```
#include<stdio.h>

int main(void)
{
    int i,j;
}
```

ただし、verbatim 環境の中で `\end{verbatim}` を表示することは出来ない。

7 図の挿入

プリアンブルに

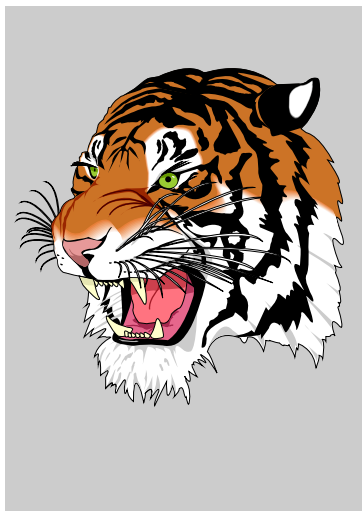
```
\usepackage{graphicx}
```

が必要。ラスター形式なら jpg, png、ベクター形式なら pdf, png が使える。ベクター形式の図が使える場合は拡大や印刷に耐えられるよう、なるべくベクター形式で貼るとよい。

Input:

```
\includegraphics[width=0.7\textwidth]{tiger.pdf}
```

Output:



`width=0.7\textwidth` は、横幅を紙の幅の 0.7 倍で表示という意味。

参考文献

- [1] 野寺隆志:「楽々 $\text{\LaTeX}$ ・第 2 版」, 共立出版 (1994).
- [2] 奥村晴彦:「[改訂第 9 版] $\text{\LaTeX}2\epsilon$ 美文書作成入門」, 技術評論社 (2023).

8 参考文献とその参照

Input:

この文章は`\cite{noder}`を参考にして
います。

現在では、`\cite{bibun}`のような文献が
よいでしょう。

```
\begin{thebibliography}{99}
```

```
\bibitem{noder}
```

野寺隆志:「楽々 $\text{\LaTeX}$ ・第 2 版」, 共立
出版 (1994).

```
\bibitem{bibun}
```

奥村晴彦:「[改訂第 7
版] $\text{\LaTeX}2\epsilon$ 美文書
作成入門」, 技術評論社 (2017).

```
\end{thebibliography}
```

Output:

この文章は [1] を参考をしています。現在
では、[2] のような文献がよいでしょう。

付録 (様々な数式記号)

ギリシャ文字	
α	<code>\alpha</code>
β	<code>\beta</code>
γ	<code>\gamma</code>
δ	<code>\delta</code>
ϵ	<code>\epsilon</code>
ε	<code>\varepsilon</code>
ζ	<code>\zeta</code>
η	<code>\eta</code>
θ	<code>\theta</code>
ϑ	<code>\vartheta</code>
ι	<code>\iota</code>
κ	<code>\kappa</code>
λ	<code>\lambda</code>
μ	<code>\mu</code>
ν	<code>\nu</code>
ξ	<code>\xi</code>
o	<code>o</code>
π	<code>\pi</code>
ϖ	<code>\varpi</code>
ρ	<code>\rho</code>
ϱ	<code>\varrho</code>
σ	<code>\sigma</code>
ς	<code>\varsigma</code>
τ	<code>\tau</code>
υ	<code>\upsilon</code>
ϕ	<code>\phi</code>
φ	<code>\varphi</code>
χ	<code>\chi</code>
ψ	<code>\psi</code>
ω	<code>\omega</code>
Γ	<code>\Gamma</code>
Δ	<code>\Delta</code>
Θ	<code>\Theta</code>
Λ	<code>\Lambda</code>
Ξ	<code>\Xi</code>
Π	<code>\Pi</code>
Σ	<code>\Sigma</code>
Υ	<code>\Upsilon</code>
Φ	<code>\Phi</code>
Ψ	<code>\Psi</code>
Ω	<code>\Omega</code>

2 項演算子	$\pm$	<code>\pm</code>
	$\mp$	<code>\mp</code>
	$\times$	<code>\times</code>
	$\div$	<code>\div</code>
	$\cdot$	<code>\cdot</code>
	$\cap$	<code>\cap</code>
	$\cup$	<code>\cup</code>
	$\oplus$	<code>\oplus</code>
	$\otimes$	<code>\otimes</code>
関係記号	$\leq$	<code>\leq</code>
	$\geq$	<code>\geq</code>
	$\neq$	<code>\neq</code>
	$\equiv$	<code>\equiv</code>
	$\sim$	<code>\sim</code>
	$\simeq$	<code>\simeq</code>
	$\approx$	<code>\approx</code>
	$\ll$	<code>\ll</code>
	$\gg$	<code>\gg</code>
	$\in$	<code>\in</code>
	$\ni$	<code>\ni</code>
	$\subset$	<code>\subset</code>
	$\subseteq$	<code>\subseteq</code>
	$\not\subset$	<code>\not\subset</code>
	$\not\subseteq$	<code>\not\subseteq</code>
矢印	$\leftarrow$	<code>\leftarrow</code>
	$\Leftrightarrow$	<code>\Leftrightarrow</code>
	$\rightarrow$	<code>\rightarrow</code>
	$\Rightarrow$	<code>\Rightarrow</code>
	$\leftrightarrow$	<code>\leftrightarrow</code>
その他の 数式記号	$\forall$	<code>\forall</code>
	$\exists$	<code>\exists</code>
	∂	<code>\partial</code>
	∞	<code>\infty</code>
	∇	<code>\nabla</code>
	$\backslash$	<code>\backslash</code>
大きさが 変化する記号	$\sum$	<code>\sum</code>
	$\prod$	<code>\prod</code>
	$\int$	<code>\int</code>
	$\oint$	<code>\oint</code>
	$\bigcap$	<code>\bigcap</code>
	$\bigcup$	<code>\bigcup</code>