

数学と情報処理 L^AT_EX

奈古屋広昭

2005 年 6 月 20 日

1 L^AT_EX について

L^AT_EX (LaTeX、らてふ、らてつく、れいてつく) は非常にポピュラーな文書処理システムです。¹ベースとなっている T_EX (TeX、てふ、てつく)² から引き継いだ

- 高品質な組版能力
- 強力な数式表現能力
- システム独立な可搬性
- 完全なプログラミング言語

といった特徴に加えて、以下のような特徴も備えています。

- 素人向け (?) の文法と命令の規則
- 論理デザイン指向
- 豊富なライブラリ

本来は欧文処理のためのシステムでしたが、有志のみなさんの尽力でほぼ完璧な日本語化がなされています³。

¹L. Lamport により 1980 年代はじめに開発された。現在普及しているのは L^AT_EX 2_ε と呼ばれるバージョン。

²D. E. Knuth により 1970 年代に開発されたとても強力な文書組版システム。ちなみにアメリカ数学会 (AMS) の標準組版システムである。

³各国/地域で韓国・朝鮮語化や中国・台湾語化、アラビア語化などもなされているそうです。

2 L^AT_EX の思想 (?)

文書を論理的な構造 (論理デザイン) と視覚的な表現 (視覚デザイン) とに分離して考えます。

文書 = 論理デザイン + 視覚デザイン

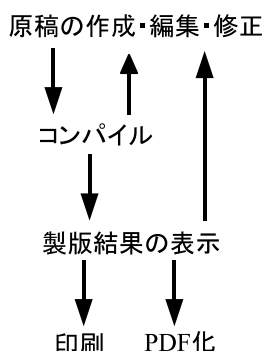
- 論理デザイン = 文章の流れ、見出し、起承転結、...
- 視覚デザイン = 用紙サイズ、フォントの大きさ、行間の幅、...

著者 (= あなた) は論理デザイン (もちろん中身) に重点を置いて文書を作成します。視覚デザインについては基本的にはデザイナーとコンピュータにまかせます。これは SGML や XML にも通じる考え方です。

誤解されないように注記しておきますが L^AT_EX では視覚デザイン (つまり見栄え) を自分で調整することができない、ということではありません。標準のデザインに不満な点があれば (L^AT_EX の機能を駆使して) 自分で調整することが可能です。実際問題として論理デザインと視覚デザインは完全に分離できるものではありません。

3 L^AT_EX での文書作成作業の流れ

L^AT_EX で文書を作成するやり方は、コンピュータのプログラムを作成することに似ています (ある観点からすると両者は同一のものとなります)。WWW ページ作成のために HTML ファイルを直接編集するやり方にも似ているかもしれません。



以下は WinShell 利用を前提としています。

3.1 原稿の作成・編集・修正

好みのテキストエディタを利用してください。WinShell 上でも可能です。なお、 \LaTeX 原稿のファイル名には “.tex” という拡張子を使うことが慣習となっています。

3.2 コンパイル

WinShell に原稿ファイルを読み込んでいない場合は、まず読み込みをおこないます。

そして「 \LaTeX の実行」をおこないます。コンパイルエラーが発生した場合は、原稿ファイルの編集へ戻ります。

3.3 製版結果の表示

「DVIVIEWの実行」により dviout を立ち上げ、画面上で製版結果を確認します。ここで修正したい部分があった場合は、原稿ファイルの編集へ戻ります。

なお dviout は再コンパイルがおこなわれたことを自動検知してくれますので、毎回「DVIVIEWの実行」をおこなわなくてもかまいません。

3.4 印刷・PDF化

必要に応じて製版結果を印刷したり PDF ファイルへ変換します。

4 \LaTeX ファイルの構造

\LaTeX は一種のプログラミング言語ですから、文法や処理方法は正確に定義されています。しかし「正確」に記述するとかなり細かい話になってしまうので、ここではごくおおざっぱに解説します（つまり以下の話には厳密にいうと間違っている説明が混っています）。

また組版命令は非常にたくさんあるので、とてもここでは紹介しきれません。以下で説明するのは、ほんのわずかの組版命令だけです。

ということで、きちんと \LaTeX を使うためには、市販の解説書を 1 冊くらいは購入した方がよいでしょう。お金がない人は図書館を利用しましょう。ある程度慣れた人ならばインターネット上の情報だけでがんばるという手もあります。

4.1 地の文と組版命令

\LaTeX のファイルはプレインテキストです。ほとんどの文字は、文字そのものとしてそのまま、つまり地の文として解釈されます。

“%” から行末（改行）までは註釈（コメント）と解釈されるので、 \LaTeX の処理系からは無視されます。

組版命令（コマンド、マクロ、コントロールシーケンス）は “\”（バックスラッシュ、環境によっては円記号 “¥” に見えることもある）ではじまる文字列です。論理的な構造（例：ここを見出しにする）や視覚的な表現（例：この文字はゴシック体で表示する）を \LaTeX の処理系に指示します。HTML のタグのようなものだといえわかりやすいかもしれません。組版命令には引数を持つものもあります。引数は { } でくくって渡します。たとえば `\underline{下線を引く}` は `\underline` という下線を引く組版命令に「下線を引く」という引数を渡したもの（ひらたくいう「下線を引く」という文章に下線を引く）と解釈されるので

下線を引く
と表示されます。

またマクロ定義という形で、新しい組版命令を自由に作ることができます。

4.1.1 空白

次の3種類の文字(空白)は基本的には無視されます。

- スペース `SPACE`
- タブ `Tab`
- 改行 

もうすこし正確にいうとこんなところです。

- 空行(2個以上連続している改行)は段落の区切りとして解釈される。
- 連続した空白はひとつの空白と同じ。
- 和文の直後の空白は無視される。
- 欧文の直後の空白は単語の境界と解釈される。

空白は「自然に」タイプしておけば「自然に」処理されます。ですから上記のことであまり頭を悩ませる必要はありません。

4.1.2 特別扱いされる文字

次の12文字は \LaTeX にとって特別な意味を持ちますので、文字そのものとしては解釈されません。これらの文字そのものを表示したいときは、その文字を表示するための組版命令を使います。

4.2 視覚デザインの指定

\LaTeX のファイルは次のような構造で書かれます。

```
\documentclass...
...                プリアンブル
\begin{document}
...                本文(実際に出力される)
\end{document}
```

文字	組版命令	意味
<code>\</code>	<code>\backslash</code>	組版命令の開始
<code>#</code>	<code>\#</code>	マクロの引数
<code>\$</code>	<code>\\$</code>	数式モードへの出入り
<code>%</code>	<code>\%</code>	註釈(コメント)の開始
<code>&</code>	<code>\&</code>	タブ制御
<code>^</code>	<code>\^{}{}</code>	上付き添字の生成
<code>_</code>	<code>_{}{}</code>	下付き添字の生成
<code>{, }</code>	<code>\{, \}</code>	グルーピング用
<code>~</code>	<code>\~{}{}</code>	単語間の空白生成
<code><, ></code>	<code>\$\$, >\$</code>	
<code> </code>	<code>\$ </code>	

そしてプリアンブルと呼ばれる部分で、文書の視覚デザインを定義する慣習になっています。

\LaTeX はドキュメントクラス(スタイルファイルともいう)とよばれる仕組みで文書のデザインを指定します。ふつうは標準で提供されている `jarticle` というクラスを使うことが多いでしょう。ドキュメントクラスの指定のために、プリアンブルに次のような組版命令を書きます。

```
\documentclass{jarticle}
```

アメリカ数学会(AMS)や電子情報通信学会(IEICE)など各種の学会では、論文投稿用に独自のドキュメントクラスを公開しています。ドキュメントクラスは自由に作成できます。

より細かい指定はオプションでおこないます。たとえば本文の文字サイズを12ポイントにし2段組みで組版したいときは次のような組版命令になります。

```
\documentclass[12pt,twocolumn]{jarticle}
```

\LaTeX はマクロという形で新しい組版命令を自由に定義することができます。マクロ定義をまとめたファイル(パッケージファイル)を読み込むためには `\usepackage` という組版命令を使います。たとえば `\keytop` という組版命令は `ascmac` というパッケージで定義されていますので

```
\documentclass[12pt,twocolumn]{jarticle}
\usepackage{ascmac}
```

とプリアンプルに書いておきます。そうすると `\keytop{あ}` という組版命令で `あ` が表示されます。

4.3 本文

4.3.1 標題

`\title` で文書のタイトルを、`\author` で著者名を、`\date` で日付を設定し、`\maketitle` で標題を出力します。

```
\title{数学と情報\LaTeX}
\author{学芸太郎}
\date{\today}
\maketitle
```

ちなみに `\date{\today}` とすると、今日の日付になります (明日や一昨日になったとしたら、あなたのコンピュータの時計は狂っています)。

4.3.2 節

`\section` で新しい節を開始します。`\subsection` で新しい小節を開始します。節番号や小節番号は自動的に計算されます。

ちなみに `\subsubsection` までは標準で定義されていますが、`\subsubsubsection` はありません。

4.3.3 段落

空行または `\\` で段落が変わります。`\LaTeX` ファイル中の空白や改行は実際の組版には影響しませんので、エディタ内での見栄えを気にする必要はありません。

4.3.4 その他

テスト用のサンプルファイル `example.tex` を参照してください。

5 実習

`example.tex` を編集して、自分で `\LaTeX` による文書を作成してみましょう。

1. 著者名を自分の名前にしてみる
2. タイトルを変更してみる

5.1 ワードプロセッサとの対比

冒頭で述べたように、`\LaTeX` は Word や一太郎に代表されるいわゆるワードプロセッサとはかなり毛色の異なるソフトウェアです。というわけでどのように使い分けるのが効果的なのか、という疑問が湧くこととなりますが、私見では次のようなかんじではないかと思います。

- 基本的には好みの問題でしょう。
- 短い文書 (1~2 ページ程度) はワードプロセッサで書いた方が楽でしょう。
- `\LaTeX` のメリットは長い論理的な文書を書くときにこそ生きてきます。
- ただし、数式が主の文章であれば短い文書でも十分メリットがあります。
- 再利用を考えると `\LaTeX` のメリットが生きてきます。
- 数式や表・図版が一切ない文書であれば、`\LaTeX` でもワードプロセッサでもなく、プレインテキストを使うのがもっとも合理的ではないでしょうか。