

# 総合図書館 TA 講座「 $\text{\LaTeX}$ で文書を作る (基礎)」

青木俊太郎 (大阪大学生命機能研究科・総合図書館)

2012-12-03

## 1 $\text{\TeX}$ / $\text{\LaTeX}$ てなに？

$\text{\TeX}$  とか  $\text{\LaTeX}$  ってなんなのさ？

$\text{\TeX}$  とは、アメリカの計算機科学者 Donald E. Knuth が 1978 年にリリースした組版処理システムのことです。 $\text{\LaTeX}$  とは、アメリカの計算機科学者 Leslie Lamport が  $\text{\TeX}$  を拡張して作った組版処理システムで、素の  $\text{\TeX}$  よりも文書作成がお手軽にできるようになっています。 $\text{\LaTeX}$  の現行バージョンが  $\text{\LaTeX 2}_\epsilon$  です。アスキーが  $\text{\TeX}$ 、 $\text{\LaTeX}$  を日本語に対応させたものを  $\text{p}\text{\TeX}$ 、 $\text{p}\text{\LaTeX}$  といいます。2012 年 12 月時点で、日本語圏で  $\text{\TeX}$  といった場合にはだいたい  $\text{p}\text{\LaTeX 2}_\epsilon$  ( $\text{\LaTeX 2}_\epsilon$  の日本語対応版) を中心とした組版処理システムのことを指していると考えていいでしょう。

組版処理システムってなんじゃい？

「 $\text{\TeX}$  は組版処理システムだ」などと言うときの「組版」とは、「本のページや表紙、レポート、パンフレット、新聞、ちらしなどなど、なんらかの紙面をデザインすること」です (本来の意味は違うけどまあ気にしない)。Microsoft Word や Adobe Illustrator なんかも組版処理システムの一種です。

Word なんかと何が違うの？

最終的に「組版された (ちゃんとデザインされた) 文書が出てくる」という点では、Word も  $\text{\TeX}$ / $\text{\LaTeX}$  も同じことをしています。しかし、Word や Illustrator などのソフトでは、組版された紙面 (ページ) のイメージがディスプレイ画面に表示され、そのイメージを直接いじって紙面をデザインします (こういうふうに、ディスプレイに表示されるものと印刷などの処理を経て出力されるものが同じになるように表現されることを、What You See Is What You Get の頭文字をとって WYSIWYG とか WYSWYG といいます)。一方  $\text{\TeX}$ / $\text{\LaTeX}$  では、ユーザが直接いじるのは文書の「ソースコード」であり、そのソースコードを変換することによって組版された文書のイメージを取得します。HTML 文書と web ページの関係に似てますね。

$\text{\TeX}$  とか  $\text{\LaTeX}$  を使うとなにがうれしいの？

元々のデータがテキストデータなので扱いが楽、数式が綺麗、ある文書のスタイル・見た目を使いまわすのが簡単、(一部の人にとっては) Word などでもちまちま作るより楽、俺コンピュータに強いぜ感を出せる、こんなところでしょうか。

$\text{\TeX}$ / $\text{\LaTeX}$  ← この変な書き方なに？

実はこれが TeX とか LaTeX の「正式な表記」なんです。

ところで  $\text{\TeX}$  ってなんて読むの？

「てふ」派と「てっく」派が主な派閥です。あんまり追究すると宗教論争になるので気にしないようにしましょう。

## 2 $\text{\TeX}/\text{\LaTeX}$ ソースコードから pdf ファイルができるまで

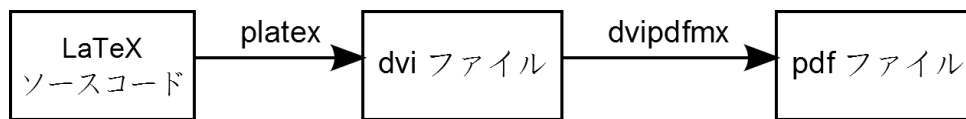


図 1  $\text{\LaTeX}$  のソースコードから pdf ファイルを作る作業の流れ。

日本語を含む  $\text{\LaTeX}$  のソースコードから pdf ファイルを作るにはふたつのステップが必要です (図 1)。まず、`platex` というプログラムで  $\text{\LaTeX}$  のソースコードを dvi ファイルに変換します。dvi (device-independent) ファイルとは、 $\text{\TeX}$  の中間出力ファイルで、デバイスに依存しないように文書の見た目を記述したファイルです (直接 dvi ファイルを扱うことはあまりないでしょうから、詳細は気にしないでください)。次に dvi ファイルを `dvi2pdf` というプログラムを使って pdf ファイルに変換します。 $\text{\TeX}$  用エディタや開発環境のなかには、これらの変換を一発でやってくれるショートカットボタンがついているものもあります。

## 3 $\text{\TeX}/\text{\LaTeX}$ のインストール

### 3.1 Windows の場合

昔は Windows 上で  $\text{\TeX}/\text{\LaTeX}$  の環境を構築するのは結構大変だったらしいですが、最近に関連するソフトウェアを自動的にインストールしてくれるインストーラがあるので、インストール自体は簡単です。2012 年 11 月時点では、`TeX` インストーラ 3 [3] を使うのがもっとも簡単でしょう。具体的なインストール方法については [4]などを参照してください。

### 3.2 Mac の場合

Mac OS X では、`MacTeX` [5] が定番です。2012 年 11 月時点、これひとつで日本語  $\text{\TeX}$  環境が整います。また、`MacPorts` などで必要なパッケージを自分でインストールしても OK です (`pTeX`, `texlive`)。

### 3.3 Linux や UNIX 系 OS の場合

パッケージマネージャで必要なパッケージをインストールします。

ソースからビルドしたい人は、..... そういう人はたぶん自力でなんとかできるよね？

## 4 $\text{\LaTeX}$ で文書をつくる

### 4.1 本文を作る前の準備

```

\documentclass[a4paper, 10pt]{jsarticle} documentclass 指定

\usepackage[top=20truemm, bottom=30truemm, left=25truemm, right=25truemm]{geometry}
\usepackage[dvipdfmx]{graphicx}
\usepackage{moreverb}
\usepackage{multicol}
プリアンプル

\renewcommand{\refname}{参考文献と情報リソース}

% 文書の情報
\title{総合図書館 TA 講座「 $\text{\LaTeX}$  で文書を作る (基礎)」}
\author{香木俊太郎 (大阪大学生命機能研究科・総合図書館)}
\date{2012-12-03}

\begin{document} document 環境

\maketitle

\section{\TeX/\LaTeX てなに?}

\begin{description}
\item[\TeX とか \LaTeX ってなんなのさ?]\mbox{()}\
  (\b \TeX) とは、アメリカの計算機科学者 Donald E. Knuth が 1978 年にリリースした組版処理シス
  (\b \LaTeX) とは、アメリカの計算機科学者 Leslie Lamport が \TeX を拡張して作った組版処理シス
  \LaTeX の現行バージョンが (\b \LaTeXe) です。

```

図 2 典型的な  $\text{\LaTeX}$  ソースコードとその構成要素。

典型的な  $\text{\LaTeX}$  のソースコードは **documentclass 指定**、**プリアンプル**、**document 環境** から構成されます (図 2)。 $\text{\LaTeX}$  ソースコードの冒頭にある `\documentclass[なんとか]{かんとか}` の部分が **documentclass 指定**です。ここで文書の種類を指定します。ソースコード中にある `\begin{document}` から `\end{document}` までの部分が **document 環境** で、ここがいわゆる文書の「本文」にあたります (ちなみに、`\begin{なんとか}` と `\end{なんとか}` で囲まれた部分を  $\text{\LaTeX}$  では**環境**といいます。頻出語なので覚えておいたほうがいいでしょう)。documentclass 指定と document 環境の間の部分を**プリアンプル**といい、各種設定やスタイルファイル、パッケージ ( $\text{\LaTeX}$  の機能を拡張するもの) の読み込み設定などをこの部分に記述します。

ちなみに、% 以降の部分はコメントで、ここの書いたものは  $\text{\LaTeX}$  の処理時には無視されます。

#### 4.1.1 documentclass の指定

documentclass 指定では、`{}` のなかで文書の種類を指定します。文書の種類によって、ページレイアウトや使えるコマンドが異なります。典型的な日本語  $\text{\LaTeX}$  システムで指定できる文書は表 1 の通りです。

report/book 系では、章分け (`\chapter`) が使えます。book では奇数ページと偶数ページのレイアウトが異なりますが (つまり、本当に「本」を想定している)、report ではすべてのページが同じレイアウトになります。授業の課題としてレポートを日本語で書く場合は jarticle か jsarticle を選べばよいでしょう。

[] の中には紙の大きさや文字サイズなどの追加設定 (ドキュメントクラスオプション) を記述します。図 2 では、用紙サイズを A4 に、文字の大きさを 12 ポイントに指定しています。省略するとデフォルト値が使用さ

れます。

表 1 日本語  $\text{\LaTeX}$  環境の主なドキュメントクラス

欧文	和文 (横書き)	和文 (縦書き)	和文 (新・横書き)	用途
article	jarticle	tarticle	jsarticle	論文・レポート
report	jreport	treport	jsreport	長い文書
book	jbook	tbook	jsbook	本

#### 4.1.2 プリアンプル

プリアンプルには文書の情報と設定、スタイルファイル・パッケージの読み込み設定などを記述します。例えば図 2 の `\title{なんとか}` では文書のタイトルを、`\author{だれだれ}` では文書の著者を指定しています。

## 4.2 タイトル

document 環境中に `\maketitle` というコマンドを書くと、そこに文書のタイトル (と著者名とか) が表示されます (サンプルはこのテキストの 1 ページ目をみて)。

## 4.3 見出し

文書の構造を明示するため節に見出しをつけるときには、以下のコマンドを使います。

```
\section{節の見出し}
\subsection{1 レベル下の節の見出し}
\subsubsection{*{2 レベル下の節の見出し}
```

`\section` だと見出しに通し番号が (自動で) つきます。一方、`\section*` のように \* がつくと、通し番号が付きません。

section の他には part で「部」の、paragraph で「段落」の見出しをつけることができます。また前述のように、report 系や book 系のドキュメントクラスを使うと chapter で「章」の見出しをつけることができます。

## 4.4 改行

段落を分けるために改行したい場合には、空行を入れます。

#### ソース

```
こういうふう  
改行しても、変換すると無視される。
```

#### 出力結果

```
こういうふう  
改行するには↑みたいに空行を入れます。
```

改行するには↑みたいに空行を入れます。

## 4.5 図を貼る

まずは貼り付ける図の画像ファイルを用意しなければなりません。扱える画像ファイルの形式は、 $\text{\LaTeX}$  ソースコードを処理するプログラムによって異なります。dvipdfmx の場合、pdf、png、jpeg、eps を使うことが

できます。

eps 以外の形式のファイルを使う場合は、画像ファイルの大きさの情報を格納したファイル (\*.xbb または \*.bb) が必要になります。開発環境によっては pdf 変換時に自動で bb ファイルを作成してくれますが、作成されない場合は extractbb というコマンドを使って自分で作ります。

また、図を貼るには、graphicx パッケージが必要です。パッケージを使うにはプリアンプルに

```
\usepackage[dvipdfmx]{graphicx}
```

と書きます。[] の中身は、 $\text{\TeX}$  ソースコードの処理に使うプログラムによって変わります。dvipdfmx で dvi から pdf に変換する場合は [dvipdfmx] としておけば OK です。

文中に図を貼るには includegraphics コマンドを使います。

```
\includegraphics[width=0.6\hsize]{latex2pdf.png}
```

{ } に貼る画像ファイルの名前を指定します。[] にはオプションを記述します。ここでは画像の幅をページの本文領域の幅の 60% に指定しています。

図を貼るだけなら includegraphics コマンドを書いておくだけでいいのですが、図の番号、説明文などを簡単に扱うためには figure 環境を使います (以下例)。

```
\begin{figure}[!h]
  \begin{center} % 図の水平位置を中央揃いにする
    \includegraphics[width=0.5\hsize]{LaTeX2pdf.png}
    \caption{\LaTeX のソースコードから PDF ファイルを作る作業の流れ。} % 図の説明文
    \label{fig:LaTeX2pdf} % 図のラベル (本文中で図を参照するときに使う)
  \end{center}
\end{figure}
```

## 4.6 表を作る

表 (table) を作るには tabular 環境を用います (ソース例中の  $\text{\verb/なんとか/}$  というのは  $\text{\TeX}$  コマンドを処理せず表示するためのコマンドで、とりあえず無視してください)。

### ソース

```
\begin{center}
\begin{tabular}{|l|c|r|}
\hline
表は & こんな感じで & つくります\\
\hline
\verb/|l|c|r| は & 各列の文字位置と & 縦線\\
\hline
左寄せ & 中央揃い & 右寄せ\\
\hline
\verb/\hline/ で & 横線を & 引きます\\
\verb/\hline/ が & ないと & ほら\\
\hline
& が & セルの & 切れ目\\
\hline
\verb/\\ / で & 次の行に & いきます\\
\hline
\end{tabular}
\end{center}
```

### 出力結果

表は	こんな感じで	つくります
l c r  は	各列の文字位置と	縦線
左寄せ	中央揃い	右寄せ
\hline で	横線を	引きます
\hline が	ないと	ほら
& が	セルの	切れ目
\\ で	次の行に	いきます

表も図と同様に `table` 環境のなかで使うと番号、説明文などを手軽に処理できます。

```
\begin{table}[!h]
\begin{center}
\caption{日本語 \LaTeX 環境の主なドキュメントクラス}
\label{tb:documentclass}
\begin{tabular}{|l|c|r|}
% 省略
\end{tabular}
\end{center}
\end{table}
```

なお、 $\text{\TeX}$  で表を作るのははっきりいって面倒くさいので、複雑な表を作りたいときは適当な支援ツールを使ったほうがいいでしょう (例えば Excel の表を  $\text{\TeX}$  に変換するツールなどがあります。詳しくは [2] の「変換ツール」のページを参照してください)。

## 4.7 数式を書く

$\text{\TeX}$  が本領発揮する数式ですが、いろんな数式の書き方を説明しているとそれだけで丸一日かかってしまうので、ここでは数式記述の基礎を説明するだけにしておきます。詳しくはググって!

文のなかに数式を書くときには、数式を書くコマンドを  $\$$  で囲みます。例えば

$\$e^{i \theta} = \cos \theta + i \sin \theta\$$  と書いておくと  $e^{i\theta} = \cos \theta + i \sin \theta$  というように出力されます (ちなみに記号やコマンドの間に空白を入れているのは筆者の好みで、空白をつめて  $\$e^{i\theta}=\cos\theta+i\sin\theta\$$  とかいても OK です)。

独立した行に数式を書くときには (これを別行立ての数式といいます)、以下のように  $\backslash[$  と  $\backslash]$  で囲った部分に数式を書きます。

```
\[
f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n
\]
```

で

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

となります。数式番号を入れたいときには `equation` 環境を使います。

## 4.8 ページの余白を設定する

素の  $\text{\LaTeX}$  で余白設定を行なうのは結構ややこしいです。具体的には、ページ上辺と下辺の余白を設定し、本文領域の高さと幅を設定します。マージンは、デフォルトのマージンである 1 インチとの差を指定します。書いているだけでめんどくさい。

最近の  $\text{\LaTeX}$  ディストリビューションにはたいてい標準で `geometry` というパッケージが含まれていると思います。これを使うとずっと簡単に余白を指定できます。具体的には

```
\usepackage[top=20truemm, bottom=30truemm, left=25truemm, right=25truemm]{geometry}
```

をプリアンプルに書いておくと、各辺の余白が指定した値に設定されます。

#### 4.9 文字の装飾・書体と大きさの変更

`\bf` `なんとか` と書くと**なんとか**と太字 (ボールド) になります。欧文だと `\it italic` で *italic* (イタリック) になります。

文字の大きさを変えるときは、表 2 のコマンドを使って標準の文字の大きさ (`\documentclass`) で指定したやつ) に対する相対的な大きさを指定する場合があります。

表 2 文字の大きさを変えるコマンド (の一部)

コマンド	見本
<code>\tiny</code> 見本 Sample	見本 Sample
<code>\footnotesize</code> 見本 Sample	見本 Sample
(標準の大きさ)	見本 Sample
<code>\large</code> 見本 Sample	見本 Sample
<code>\Large</code> 見本 Sample	見本 Sample
<code>\LARGE</code> 見本 Sample	見本 Sample
<code>\huge</code> 見本 Sample	見本 Sample
<code>\Huge</code> 見本 Sample	見本 Sample

書体を変えるときは、例えば和文で `\textgt{ゴシック体}` とするとゴシック体になります。

## 参考文献と情報リソース

- [1] 奥村晴彦『改訂第5版  $\text{\LaTeX}2e$  美文書作成入門』技術評論社、2010
  - [2] TeX Wiki <http://oku.edu.mie-u.ac.jp/~okumura/texwiki/> (2012-11-19 閲覧)
  - [3] TeX インストーラ 3 <http://www.math.sci.hokudai.ac.jp/~abenori/soft/abtexinst.html> (2012-11-19 閲覧)
  - [4] もう難しいなんて言わせない! たった 20 分でできる簡単 TeX インストール Windows 編 (2012 年 4 月版) — 情報科学屋さんを目指す人のメモ <http://did2memo.net/2012/04/23/easy-latex-install-windows-201204/> (2012-11-19 閲覧)
  - [5] MacTeX - TeX Users Group <http://www.tug.org/mactex/> (2012-12-03 閲覧)
- [1] は  $\text{\TeX}/\text{\LaTeX}$  を使うなら必携の一冊。わからないことがあったら、まずはこの本にあたってみよう。[2] には  $\text{\TeX}/\text{\LaTeX}$  の情報が豊富にまとまっている。環境構築や関連ソフトウェアに関する情報も多い。