



Title	日本語TEXの使い方
Author(s)	上原, 龍也
Citation	大阪大学大型計算機センターニュース. 1988, 71, p. 57-66
Version Type	VoR
URL	https://hdl.handle.net/11094/65805
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の使い方

上原 龍也

大阪大学 産業科学研究所

1 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ とは ?

最近ワープロが普及し、手書きに変わって文書をプリンタで出力する機会が増えてきた。しかし、ワープロのプリンタ出力では不満がある人が少なくないと思う。ワープロの性能もだんだんと良くなって来ているが、字体や文字のサイズや書式を自由に換えられる訳ではない。これらの要望に答えることができるソフトウェアに $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ がある。 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ [2] とは、スタンフォード大学の Knuth 教授が作成したソフトウェアで、コンピュータを使って印刷のような高品位の文書を生成する清書プログラム (デスクトップパブリッシングともいう) である。

$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ は、ワープロのように画面上でレイアウトを決めていくのではない。 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ では、まずエディタなどを用いて文章を入力する。字体やレイアウトの指定は、すべて文章ファイルにコマンドを挿入することにより行なわれる。そして、この文書ファイルを $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ で処理することにより、プリンタに出力することができる。したがって、文書ファイルのままではレイアウトがどうなっているか分からない。このレイアウトの確認をするために、出力イメージを画面上に表示することもできる。

$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の特長の一つとして、さまざまな文章の構造を扱えることがある。たとえば、論文を執筆する場合を考えると、タイトル、著者名があつて、内容梗概、本文、そして参考文献がくる。また、本文はいくつかの章からなっており、その文中には表や図などが挿入されている。 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ では、これらの構造をコマンドによって表現し、それぞれの書式にしたがつて出力を整える。このとき、章、節、図などのナンバリングやその参照は自動的に行なってくれるので、節の追加や図の移動も簡単にできる。さらに、これらのコマンド自体の再定義も可能なので、自分なりの書式を作ることにもできるという利点がある。

もうひとつの $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の特徴として、強力な数式の表示能力がある。これは、 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ が数式

の処理を第一の目的として設計されているからである。たとえば、複数行にわたるような積分式や行列、分数式などもいくつかのコマンドを使うことによって容易に記述できる。もちろん、 ∂ や \Re のような数学に特有な文字や、 \bar{a} や \tilde{a} などの表示も簡単である。

ここでは、アスキーの倉沢良一氏が $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ を日本語用に拡張した‘日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ’、特にそのマクロパッケージである $\mathrm{J}\mathrm{I}\mathrm{A}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の使い方について説明する。 $\mathrm{J}\mathrm{I}\mathrm{A}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ は、日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ のマクロ機能を利用して、日本語 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の持つ機能を簡単に利用できるようにしたものである。以下では、この $\mathrm{J}\mathrm{I}\mathrm{A}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ を用いた文書ファイルの作り方について説明する。なお、この文章自体も $\mathrm{J}\mathrm{I}\mathrm{A}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の出力例となっている。

2 文字の表現

文章の中にはさまざまな文字が含まれている。これらの文字のうち、英数字や漢字は入力された文字がそのまま出力される。ただし、英字については文字の幅に合わせた割付、単語の間隔の調節、ハイフンの挿入による右ぞろえなどが自動的に行なわれる。たとえば、

```
With modern computers, typesetting is not just for books and  
documents aimed at a wide audience.
```

と入力すると、

```
With modern computers, typesetting is not just for books and documents  
aimed at a wide audience.
```

と印刷される。ここで注意しなければならないことは、空白と改行の扱いである。空白はいくつ続いていても1つの空白と見なされる。もし、実際に複数の空白を使いたければ、`\hspace`を使えばよい。また、改行は空白と同じものとして扱われる。したがって、改行を挿入しても実際の出力上では改行されない。連続した改行によって空行があれば、そこが段落の終わりと見なされる。

一方、英数字以外の記号、たとえば、‘!’、‘.’、‘;’のような記号は下の特別な意味を持つ10文字を除いてそのまま入力できる。

\$ % & ~ _ ^ \ { }

これらの文字のうち# \$ % & _ { }を印刷するには直前に \ を挿入すればよい。(例 : \#) さらにキーボードにない記号は ' ' で始まるコマンドを使う。、\$は\\$、ℓは\pounds として表わされる。他に注意しなければならないのは、クォテーションである。左のシングルクォテーションは‘で、右は’ で表わす。ダブルクォテーションは"ではなく、左右それぞれ‘ ‘と’ ’ を用いる。

また、字体もコマンドで指定する。このコマンドと字体の関係を表 1 に示す。ただし、漢字については現在のところ、明朝体¹とゴシック体のみである。もし、特定の部分だけ字体を変えたいならば、{\gt ...}と書けばよい。この {} で囲むことをグループ化といい、コマンドはこの中のみで有効である。

コマンド	字体	例
\rm	ローマン	Roman
\it	イタリック	<i>Italic</i>
\bf	ボールド	Bold
\sl	斜体	<i>Slanted</i>
\sf	サンセリフ	sans serif
\min	明朝体	明朝体
\gt	ゴシック体	ゴシック体

表 1: 字体を指定するコマンド

文字の大きさもコマンドで指定する。図 2 はコマンド名と文字の大きさの関係を表わしたものである。たとえば、{\footnotesize 漢字} とすれば ‘漢字’、{\Large 漢字} とすれば ‘漢字’ となる。

3 文章の構造

3.1 文章のスタイル

前述したように、文章にはさまざまな構造があるが、文章の種類によってそれぞれスタイルが異なっている。J_AT_EX には、このような構造に対して、文書スタイルをいくつか用意しており、それぞれのスタイルで清書することができる。標準で用意されているスタイルには次のものがある。

¹J_AT_EX では、\min が実際はマイナス文字 (−) と解釈されてしまう。 \min のかわりにコマンド \pmin を使えば正常に動作するようだ。

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

表 2: 文字の大きさ

- `jarticle` 論文の形式
- `jreport` テクニカクレポートのような形式
- `jbook` 一冊の本の形式
- `letter` (英語の) 手紙の形式²

また、他にさまざまなスタイルも配布されており、また自分のスタイルを作ること
 できる。

3.2 文章構造の入力

ここでは、論文を作成するための文書ファイルの書き方について説明する。図 2 の
 ような出力を得るための文書ファイルを図 1 に示す。

まず、ファイルの先頭には

```
\documentstyle[options]{style}
```

が必要である。`style` には先ほどのスタイル名が入る。`options` には、11pt や 12pt のフォ
 ントサイズ (デフォルトは 10pt) や、二段組を指定する `twocolumn` などがある。

次の行の `\begin{document}` は、最後の行の `\end{document}` と対になっており、両者
 に囲まれた部分が本体となる。このように、 \LaTeX では `\begin{name}` と `\end{name}`
 で囲まれた部分を といい、この環境のみで、設定されているコマンドが使えるよう
 になる。その次の 4 行はタイトルを作るためのコマンドである。`\title` は論文の題を、

²手紙の形式は、英語と日本語では異なるので、日本語版は作られていないようです。

```

\documentstyle[twocolumn,11pt]{jarticle}
\begin{document}
\title{題名}
\author{著者の名前}
\date{昭和 63 年 1 月 1 日}
\maketitle
\section{この章の名前}
\subsection{この節の名前}
    この節の本文をここに書く。
\subsection{この節の名前}
    最近ワープロが普及し、手書きに変わって文書をプリンタで ...
\end{document}

```

図 1: 図 2 のための入力ファイル

`\author` は著者名を、`\date` は日付を指定するコマンドである。そして、`\maketitle` によって初めてタイトルが生成される。

題名	
著者の名前	
昭和 63 年 1 月 1 日	
1 この章の名前	さる。TeX とは、コンピュータを使って、
1.1 この節の名前	印刷のような高品質の文書を生成する清書
この節の本文をここに書く。	プログラムである。
1.2 この節の名前	TeX は、ワープロのように画面上でレイアウトを決めていくのではない。TeX で
最近ワープロが普及し、手書きに変わって	は、まづエディタなどを用いて文書を入力
で文書をプリンタで出力する機会が増えて	する。字体やレイアウトを指定は、すべて
きた。しかし、ワープロのプリンタ出力で	文章ファイルにコマンドを挿入すること
は不満を持つ人が少なくないと思う。ワ	より行なわれる。そして、この文書ファイ
プロの性能もだんだんと良くなって来	ルを TeX で処理することにより、プリン
らるが、字体や文字のサイズや書式を自由	に出力することができる。したがって、
変えられる訳ではない。これに対して、TeX	文章ファイルのままだではレイアウトがどう
を用いればこれらの要望に答えることが	なっているか分からない。このレイアウト

1

図 2: 論文の形式

実際の本文は、`\section` から始まる。本文の章立ては、

`\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\subparagraph`

の順で細くなる。各章の題は、`{ }` で挟んでコマンドの次に書けばよい。なお、章の番号は自動的に付けられる。

1. 1 番目の項目	<code>\begin{enumerate}</code>
2. 2 番目の項目	<code>\item 1 番目の項目</code>
(a) 入れ子になった項目	<code>\item 2 番目の項目</code>
• 項目その 1	<code>\begin{enumerate}</code>
• 項目その 2	<code>\item 入れ子になった項目</code>
	<code>\end{enumerate}</code>
	<code>\end{enumerate}</code>
	<code>\begin{itemize}</code>
	<code>\item 項目その 1</code>
	<code>\item 項目その 2</code>
	<code>\end{itemize}</code>

図 3: 箇条書きのコマンドとその実行例

また、箇条書きのためのコマンドも用意されている。(図 3 を参照) `enumerate` 環境を用いれば、書く項目ごとに自動的に番号を付けてくれる。また、`itemize` 環境は、番号をつけずに項目を列挙するものである。これらの環境は、入れ子にすることも可能である。

3.3 相互参照

論文では、本文中で図や表の番号、あるいは参考文献などを参照することがよくある。しかしながら、これを手作業で行なうのはたいへんであり、また、挿入などがあると最初からやり直さなければならない。このため、 $\text{JAT}_{\text{E}}\text{X}$ ではこのような相互参照を自動的行なう機能がある。たとえば、図の領域を確保するには、次のような `figure` 環境を用いる。

```
\begin{figure}
\vspace{8.5cm}
\caption{図の題名}
\label{example}
\end{figure}
```

`\vspace` は図の高さ、`\caption` は図の題名を指示している。`\label` は参照のために使われるラベルを表わしてゐる。本文中で '`\ref{example}`' があると自動的にコマンドの部分が '`example`' というラベルをもつ図に割り立てられた番号に置き換えられる。

また、参考文献の参照についてもサポートしている。本文中で参照したい部分に '`\cite{label}`' をおくと、そのラベルに対応する参照番号で置き換えられる。文献データは、次のように書く。

```

\begin{thebibliography}{99}
\bibitem[Name 86]{label1} 文献 1
\bibitem[Who 85]{label2} 文献 2
\bibitem[You 82]{label3} 文献 3
\end{thebibliography}

```

パラメータ“99”は、参照時に使われるラベルの最大長を表わしている。この場合 2 桁なので、ラベルの数字は 1～99 の範囲である。`\bibitem` の後の `[]` で囲まれた語は、文献一覧を印刷するときの見出し語となる。

4 数式の表示

$\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の数式の表示能力は非常に優れている。これは、AMS (アメリカ数学会) が論文を $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ の出力で受け取ることからも分かる。ここでは、豊富な機能をすべて紹介する訳にはいかないので、ほんの一部のみを紹介する。詳しくは文献 [1] [2] を参照のこと。

数式を書くには、`\[と\]` で囲む必要がある³。たとえば、分数式はコマンド

```
\frac{分子}{分母}
```

を用いればよい。なお、以下では左のコマンド列が右のように出力されることを示している。

$$\frac{y+z}{y^2+1} \quad \backslash[\backslashfrac{y+z}{y^2+1} \backslash]$$

この記述の中の‘`^`’は上付き文字を、また‘`_`’は下付き文字を指定している。このような機能を利用すればもっと複雑な式も書くことができる。

$$\sum_{i=1}^n x_i = \int_0^1 f \quad \backslash[\backslashsum_{i=1}^n x_{i} = \backslashint_{0}^1 f \backslash]$$

さらに、`array` 環境を用れば行列も容易に表現できる。

$$\left(\begin{array}{cc} \cos \frac{\sqrt{3}}{2} & \sin \theta \\ \ddot{a} & x+y \end{array} \right) \quad \backslash[\backslashleft(\backslashbegin{array}{cc} \backslashcos{\backslashfrac{\sqrt{3}}{2}} & \backslashsin\theta \\ \backslash\ddot{a} & \backslashx+y \backslashend{array} \backslashright) \backslash]$$

³行の中に挿入する場合には、`$$`で囲む方法がある

これらのコマンドを駆使すれば、さらに複雑な数式でも記述することができる。

5 \TeX の使い方

以上に述べたようなコマンドを使って文書ファイルを作るわけであるが、この際、自分の使い馴れたエディタを使えば良いし、また端末にグラフィック機能の必要もない。これは、清書システムの大きな特徴でもある。なお、文章のファイル名には、拡張子として “.tex” を付けておくほうがよい。次に、この文章ファイルを jlatex で処理すると、拡張子が “.dvi” のファイル (dvi ファイルと呼ぶ) ができる。この dvi ファイルは出力デバイスに依存しない形式で書かれているので、出力のためのドライバがあれば、さまざまな出力機器を使うことができる。たとえば、私の研究室ではキャノンのレーザビームプリンタ LBP-8 II AJ2、ソニーのレーザビームプリンタ NWP-533 などに出力することができるようになっている。また、同じファイルは画面にも出力することができる。これをプレビューアという。この例を図 4 に示す。(これは、ウィンドウシステム SunView 上のプレビューア⁴である。そのほかに X ウィンドウ用⁵がある。) プレビューアを使えば、紙に出力しなくともレイアウトの確認ができる。

なお、プリンタやプレビューアのコマンドは、システムによって異なっている。近いうちに、大型計算機センターの SUN にも \TeX がインストールされるので、そのときには分かるでしょう。

6 おわりに

JL \TeX について説明してきたが、紙面の都合で説明できなかった機能がいくつある。

- picture 環境 — 簡単な図を書くことができる。
- tabular 環境 — 表を書くための環境。
- マクロの使い方 — 自分でコマンドをできる。
- ディスプレイや脚注などのさまざまな書式のための環境

⁴このプレビューアは、junet のニュースに流れていた pds (パブリックドメインソフトウェア) である Neil Hunt の dvipage を私が日本語 \TeX 用に改造したものである。近日中に、pds として公開する予定。

⁵このプレビューアは産業科学研究所の川口敦生氏が作成したもので、すでに pds として配布されている。

など、さまざまな機能がある。また、本文で説明したものでも細かな点については省略しているので、これらのことについては \LaTeX のマニュアル[1]を参照すればよい。さらに、 \TeX についての説明は、“ \TeX Book”[2]に書かれている。また、*bit*の特集[3]も役に立つでしょう。

まず、簡単な文章でも良いから、一度使ってみる事です。その出力の美しさには感激するでしょう。

謝辞

\TeX や \LaTeX についていろいろと御教授頂き、また、本稿作成に御協力いただいた大阪大学産業科学研究所 上原邦昭講師に深く感謝の意を表します。

参考文献

- [1] Leslie Lamport : “ \LaTeX : *A Document Preparation System*”, Addison-Wesley, 1986.
- [2] Donald E. Knuth : “ \TeX book”, Addison-Wesley, 1984.
- [3] 大野義夫他 : “ \TeX 入門”, *bit*, Vol.19 No.7 ~ Vol.20 No.5, 1987 ~ 1988.

