

第3回 L^AT_EX ゼミ

ゼミ担当者 : 降幡 建太郎, 澤田 淳二, 谷口 義樹
 指導院生 : 水田 伯典, 小椋 信弥
 開催日 : 2002年4月20日

ゼミ内容: 本ゼミでは, T_EX で表の基本的な作成方法と, dvi ファイルの PDF 化, スタイルファイルの説明を行います. また, mikilab スタイルの使用方法についても説明します.

1 表の作成

図版と同様, L^AT_EX には表を最適な場所に配置するための環境も用意されています. 本章では, 表を配置したり作成したりする環境について解説します.

1.1 表組みの基本

最も基本的な表組みの例を Table 1.1 に示します.

Table 1 最も基本的な表

書名	数量	金額
進化する人工物	1	1400 円 (税別)
未来革命	1	1700 円 (税別)

上の表組みを出力するソースは Fig. 1 のとおりです.

```

\begin{center}
\begin{tabular}{lcr}
書名 & 数量 & 金額 \\
進化する人工物 & 1 & 1400 円 (税別) \\
未来革命 & 1 & 1700 円 (税別) \\
\end{tabular}
\end{center}
```

Fig. 1 最も基本的な表のソース

center 環境は表組みと直接関係ありませんが, center 環境がなかったら, 表は別行立てにならず, 本文の中に入り込んでしまいます. これは, 表は一つの大きい文字として扱われているためです.

`\begin{tabular}...`までが表そのものを出力する tabular 環境です. tabular 環境の書式と引数の指定をそれぞれ, Fig. 2 および Table 2 にまとめておきます.

引数 (列指定) は列の数だけ並べます. Table 1.1 では列指定は lcr でしたので, 1 列目は左寄せ, 2 列目は中央, 3 列目は右寄せになっています.

列の区切りは &, 行の区切りは \\ です.

```

\begin{tabular}{引数 (列指定)}
:
\end{tabular}
```

Fig. 2 tabular 環境の書式

Table 2 tabular 環境の引数

指定	解説
l	左寄せ
c	中央
r	右寄せ
	縦の罫線
	縦の二重罫線

1.2 罫線

1.2.1 罫線の引き方

前節で表組みの作成を解説しましたが, この節では表にはなくてはならない罫線の引き方を説明します. 実際に前節で作成した表組みに罫線を加えたものを Table 3 に, そのソースを Fig. 3 に示します.

Table 3 罫線を引いた表

書名	数量	金額
進化する人工物	1	1400 円 (税別)
未来革命	1	1700 円 (税別)

縦罫線の引き方は { 列指定 } の中の該当個所に半角縦線 | を入れるだけです. また 2 重の縦罫線にしたいときは || と書きます.

横罫線は \hline です. \hline \hline と書くと, 2 重の横罫線になります.

1.2.2 部分的に罫線を引く方法

\hline 命令を使用すると, 表の幅の文だけ横の罫線を出力します. しかし, 場合によっては, 1, 2, 4 列目

```
\begin{tabular}{|l|c|r|} \hline
書名 & 数量 & 金額 \\ \hline \hline
進化する人工物 & 1 & 1400 円 (税別) \\ \hline
未来革命 & 1 & 1700 円 (税別) \\ \hline
\end{tabular}
```

Fig. 3 罫線を引いた表のソース

だけに引きたい場合があります。例えば、Table 4 のような表を作りたいとします。

Table 4 部分的に罫線を引いた表

出版社	書名	著者	定価
A 社	TeX とは何か	鈴木一郎	1400 円
B 社	TeX の活用術		1800 円

```
\begin{tabular}{|c|c|c|c|} \hline
出版社 & 書名 & 著者 & 定価 \\ \hline \hline
A 社 & \TeX とは何か & 鈴木一郎 & 1400 円 \\ \cline{1-2}\cline{4-4}
B 社 & \TeX の活用術 & & 1800 円 \\ \hline
\end{tabular}
```

Fig. 4 部分的に罫線を引いた表のソース

Fig. 4 に示したように、この場合には `\hline` 命令の代わりに `\cline` 命令を使います。

`\cline` 命令は `\cline{数字 - 数字}` のように罫線を引くはじめたい列から引き終わりたい列までの数字を書きます。

1.3 1行あたりの要素数の変更

表を組んでいると、たとえば要素数は4つだが、見出しの行だけ4つの要素の欄をまとめて使用したいとか、表の要素は左揃えで出力したいが、見出しの要素だけは中央揃えにしたいということがあります。たとえば、Table 5 のように出力したいという場合です。

Table 5 は「請求書」が3列分まとめて中央揃えで出力されているのがわかります。Table 5 のソースは Fig. 5 のようになります。

列をまとめるには、`\multicolumn` という命令を用います。この `\multicolumn` 命令は3つの引数を持っており、前から順にまとめる欄の数、まとめた欄における要素の配置位置、要素となっています。Fig. 5 では、

```
\multicolumn{3}{|c|}{\textbf{請求書}} \\ \hline \hline
```

Table 5 1行目をまとめた表

請求書		
書名	数量	金額
進化する人工物	1	1400 円 (税別)
未来革命	1	1700 円 (税別)

```
\begin{tabular}{|l|c|r|}
\multicolumn{3}{|c|}{\textbf{請求書}} \\ \hline \hline
\multicolumn{1}{|c|}{書名} & 数量 & 金額 \\ \hline \hline
進化する人工物 & 1 & 1400 円 (税別) \\ \hline
未来革命 & 1 & 1700 円 (税別) \\ \hline
\end{tabular}
```

Fig. 5 1行目をまとめた表のソース

となっていますが、この行は次のような意図によって指定されています。

1. `tabular` 環境で作成されるすべての欄をまとめて見出しにしたい

Fig. 5 の `tabular` 環境で作成される欄の総数は3つです。そこでまず、`\multicolumn` 命令で最初の欄以降の3つの欄をまとめます (`\multicolumn` 命令の第1引数を "3" にします)。

2. 見出しは中央揃えにしたい

`\multicolumn` 命令でまとめた3つの欄を、新たに要素数が1つで、要素を中央揃えで配置する欄として定義しておきます。なお、要素の左右には縦罫をひきます (`\multicolumn` 命令の第2引数を "|c|" にします)。

3. 見出しの欄には、太字で請求書と出力したい

`\multicolumn` 命令で新たに設定された欄に、"請求書" という文字列を入れます (`\multicolumn` 命令の第3引数を "{\textbf{請求書}}" にします)。

1.4 横幅の決まった表

全体の横幅の決まった表は `tabular` 環境の代わりに `tabular*` 環境を使います。Fig. 6 のように使います。

幅を合わせるために列間に均等に空きが入ります。Table 6, Fig. 7 は幅を 80mm にしたときの表と、そのソースです。

Fig. 7 の列指定において、`p{10zw}` は「全角 10 文字幅」という意味で、入りきらなければ次行送りになりま

```
\begin{tabular*}{(幅)}{@{\extra
colsep{\fill}}(列指定)}
:
\end{tabular}
```

Fig. 6 横幅を決める表の書式

Table 6 横幅を 80mm にした表

請求書		
書名	数量	金額
進化する人工物	1	1400 円 (税別)
未来革命	1	1700 円 (税別)

す。

1.5 table 環境

これまで、tabular 環境の解説をしてきましたが、tabular 環境はあくまで表を作るだけですので、表の上部に説明を加えたり、本文中でその表を参照するなどということはできません。そこで用いるのが、表全体の領域を確保するための table 環境です。table 環境を用いれば、figure 環境のように中央揃えや、

```
\caption{説明}
```

で表の説明をつけたり、

```
\label{参照名}
```

で、参照名をつけたりできます。一般的に表の説明は表の上に来るように書きます。

後で説明するように、mikalab スタイルでは本文中での表の参照は \tbref を用います。たとえば、

```
… については、\fbref{table1}を参照のこと。
```

```
\begin{tabular*}{80mm}{@{\extracolsep{\fill}}|p{10zw}|r|r|}\hline
\multicolumn{3}{|c|}{\textbf{請求書}} \\\hline\hline
書名 & 数量 & 金額 \\\hline
進化する人工物 & 1 & 1400 円 (税別)\\
未来革命 & 1 & 1700 円 (税別)\\ \hline
\end{tabular}
```

Fig. 7 横幅を 80mm にしたときの表のソース

とソースに記述しておく、出力結果は

```
… については、Table 2 を参照のこと。
```

のようになります。

2 DVI ファイルの PDF 化

2.1 何故、PDF を用いるのか

PDF¹ファイルとは、Adobe System が開発した電子文書の形式です。

一般に、あるソフトを用いて作成された文書を表示するには、相手のコンピュータにも同じソフトがインストールされていないといけません。たとえば、TeX で作成された文書を見るには、DVI ドライバがインストールされている必要があります。また、同様に Microsoft Word で作成された文書を見るには、Word がインストールされていないと見ることはできません。これでは、インターネット上で不特定多数の人に自分の書いた文書を見てもらうのに障害となってしまいます。

そこで、作成した文書を PDF 形式に変換します。文書を PDF 形式に変換しておけば、相手のコンピュータには PDF を表示するプログラムだけ入っていればよく、文書別の表示ソフトは必要なくなります。

PDF 形式を用いることで、次のような利点があります。

- 相手のコンピュータの機種や環境によらず、元の文書と同様のレイアウトで文書を表示することが可能
- 閲覧用のソフトは無償で配布されている
- プラグインを用いることによって、Web ブラウザ上でも閲覧が可能

2.2 PDF 化のための準備

PDF への変換を行うには、Adobe Acrobat 5.0 が必要になりますので、まず、各自でインストールを行います。

インストールが無事に完了すると、Acrobat と一緒に Acrobat Distiller 5.0 (以下 Distiller) がインストールされているはずですが、この Distiller で最終的な PDF への変換を行うのですが、初期設定のままではうまく変換することができません。したがって、まず Distiller を三木研用にカスタマイズする必要があります。これは、三木研用の設定が保存された設定ファイルが存在しますのでこれを利用します。設定ファイルは、PrintOptimized_mikalab.JobOptions という名前以下のディレクトリに置いてあります。

```
¥¥museion ¥ archive ¥ isdl ¥ basic_seminar
¥ TEX ゼミ ¥ Distiller
```

¹Portable Document Format

このファイルを Distiller の他の設定ファイルとともに格納しておきます。設定ファイルは、デフォルトでインストールした場合は次のディレクトリに格納されています。

C:\Program Files\Adobe\Acrobat 5.0\Distiller\Settings

Distiller を起動すると、Fig. 8 のようなウィンドウが開きますので、ジョブオプションの欄を PrintOptimized_mikilab にします。

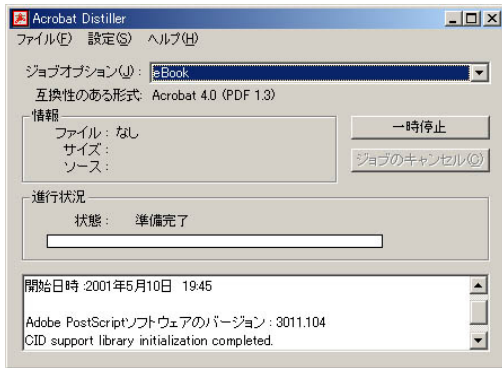


Fig. 8 Distiller の設定

2.3 PDF への変換

次に実際に PDF への変換を行います。DVI ファイルから直接 PDF へと変換することはできません。したがって、まず始めに \TeX のソースファイルから PS ファイルを生成し、それをさらに PDF へ変換するという方法をとります。

• PS ファイルの作成

秀丸において、 \TeX ソースをコンパイルするためのマクロを起動した際、”PS ファイルに変換して表示”を選択します (Fig. 9 参照)。 \TeX ソースにエラーがなければ、PS ファイルが生成されます。

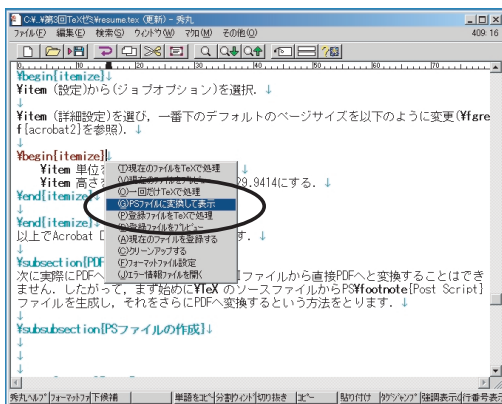


Fig. 9 秀丸のマクロ

• PDF への変換

Distiller を起動します。Fig. 8 のウィンドウが開きますので、生成した PS ファイルをこのウィンドウへドラッグ & ドロップします。特に問題がなければ、これで PDF ファイルが生成されます。

3 スタイルファイル

3.1 スタイルファイルとは？

$\LaTeX 2_{\epsilon}$ では、文書ファイルには文書構造だけを指定し、文書構造と実際のレイアウトとの対応は、クラスファイル (cls ファイル) という別ファイルで定義するという方法をとっています。また、それらクラスファイルを補う形で、いろいろなスタイルファイル (sty ファイル) を必要に応じて文書ファイルのプリアンブルで読み込ませます。

3.1.1 mikilab.sty について

月例発表会スタイルファイル (mikilab.sty) は、知的システムデザイン研究室の月例発表会のレジメ作成用のスタイルオプションファイルで、そのためのマクロがいくつか登録されています。今回の \TeX ゼミでは、それらマクロの使い方を解説します。

mikilab スタイルを用いるには、原稿のプリアンブルに次のように記述します。

```
\usepackage{mikilab}
```

3.1.2 mikilab.sty のマクロ

mikilab.sty には、月例発表会のレジメ用のマクロがいくつか登録されていますが、具体的には以下のとおりです。尚、各マクロ命令の詳しい説明は、第 3 章のコマンド集にありますのでそちらを参照してください。

• ヘッダ

1 ページ目のヘッダに月例発表会の通算回数と開催年・月を記述するための、`\beginheader` というマクロ命令が用意されています。

• 題目

日本語題目を入力するために `\title` というマクロ命令が用意されています。題目を出力するのに `\maketitle` は必要ありません。

• 著者名

著者名を入力するために `\author` というマクロ命令が用意されています。連名の場合には「」で区切ってください。著者名を出力するのに `\maketitle` は必要ありません。

以上のマクロ命令を用いると、Fig. 8 のような出力が得られます。

月例発表会レジュメ用スタイルファイル (mikilab.sty) の使い方
How to use mikilab.sty学生 氏名
Shimei GAKUSEI

Fig. 10 ヘッダ, 題目および著者名

● 概要

レジュメの概要を英文で 4 行程度にまとめてください。概要の記述には abstract 環境を利用します。出力は、たとえば次のようになります。尚、太字の Abstract: は自動的に加えられます。

Abstract: This is a sample document which uses the monthly lecture meeting at Mikilab. This is a sample document which uses the monthly lecture meeting at Mikilab. This is a sample document which uses the monthly lecture meeting at Mikilab. This is a sample document which uses the monthly lecture meeting at Mikilab. This is a sample document which uses the monthly lecture meeting at Mikilab. And so on...

Fig. 11 概要の出力

● 図, 表および式の参照

本文中での図, 表, 式の参照はすべて \ref 命令を用いて行ってきましたが, mikilab スタイルでは新たなマクロが登録されています。図を参照するには \figref, 表を参照するには \tbref, 数式を参照するには \eqref を用いてください。

4 mikilab スタイルで使用できるコマンド一覧表

4.1 先頭に関するコマンド

<code>\documentclass</code>	<p>mikilab スタイルでは、デフォルトで<code>\documentclass[a4paper,10pt]{jarticle}</code>と表示されます。[] の中で、用紙の大きさ、基本となる文字の大きさ、用紙の方向、段組み、印刷の種類、標題・概要ページの扱い、試し刷り、章の開始ページの設定などを決めることができます。</p> <p>次に {} の中では、文書クラスごとのデフォルトのクラスオプションを設定できます。今回の jarticle の場合では、A 4 用紙・標準文字サイズ 10pt・片面印刷・1 段組み・試し刷りの設定 OFF というデフォルトになっています。また、jreport に変更すると、章 (section) が使用可能になります。</p>
-----------------------------	--

<code>\usepackage</code>	<p>L^AT_EX は、新たなマクロを定義した「パッケージ」と呼ばれるファイルを読み込むことで、命令を変更したり追加したりすることができます。いくつかのパッケージは標準の L^AT_EX の命令が持っている制限事項を取り払い、またあるパッケージは L^AT_EX が使用できるフォントの数を増やし、別のパッケージはたくさんの命令や環境を追加します。</p> <p>つまり、パッケージを読み込むことで、L^AT_EX に表現可能な文書の世界を大きく広げてあげるのです。パッケージは通常、拡張子が“.sty”のファイルによって提供されます。</p> <p>三木研究室では、mikilab.sty を使うことにより、すべての環境が整います。</p>
--------------------------	--

<code>\onecolumn</code>	新規ページを起こして、以降を一段組みとします。
<code>\twocolumn</code>	新規ページを起こして、以降を二段組みとします。

具体例：`abstract` 環境を二段抜きで出力する場合

<pre> \documentclass[a4j,twocolumn]{jarticle} \usepackage{mikilab} \pagestyle{empty} \begin{document} \twocolumn[% \begin{center} {\LARGE\textbf{二段抜き}}\ \vspace{10mm} {\large\textbf{abstract 部}}\ \vspace{3mm} この部分は二段抜きされている部分です \vspace{3mm} \end{center}} この部分から 2 段組が始まります。 \newpage 2 段組の右側です \end{document} \end{tabular} \end{minipage} </pre>

Fig. 12 ソースファイル

<p>二段抜き</p> <p>abst t部</p> <p>この部分は二段抜きされている部分です。</p> <p>この部分から 2 段組が始まります。 2 段組の右側です。</p>

Fig. 13 Fig. 12 の実行結果

<code>\pagestyle</code>	ページ番号を出力するのか、そしてどこに出力するのかをプリアンブルで設定することができます。引数が <code>plain</code> なら、各ページの下部にページ番号を出力し、 <code>empty</code> なら、ページ番号を一切出力しません。
<code>\setcounter{page}{0}</code>	ページ番号の初期値を変更します。これでは0ページからになります。
<code>\thispagestyle</code>	特定のページだけページ番号の出力方式を変えたい場合に、そのページの適当な部分に記述します。引数は <code>\pagestyle</code> と同じです。

4.2 月例用ヘッダに関するコマンド

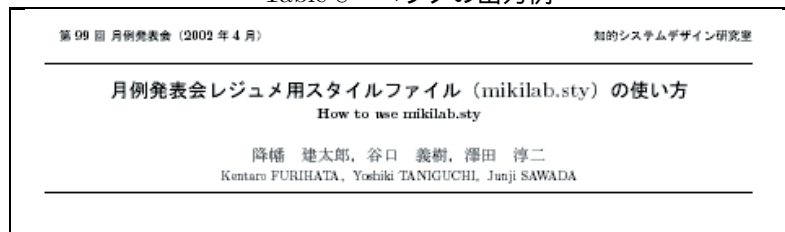
<code>\beginheader</code>	1 ページ目のヘッダに月例発表会の通算回数と開催年・月を記述するための、 <code>\beginheader</code> というマクロ命令を用意しました。このファイルのソースを参照し、以下のような書式で題目を入力してください。
<code>\title</code>	題目を入力するために <code>\title</code> というマクロ命令を用意しました。このファイルのソースを参照し、Fig. 7 のような書式で題目を入力してください。
<code>\author</code>	著者名を入力するために <code>\author</code> というマクロ命令を用意しました。このファイルのソースを参照し、Fig. 7 のような書式で題目を入力してください。

ヘッダの作成例

Table 7 ヘッダのソース

```
\begin{document}
\beginheader{99}{2001}{4}
\title%
{月例発表会レジュメ用スタイルファイル (mikilab.sty) の使い方}%
{How to use mikilab.sty}
\author{降幡 建太郎, 谷口 義樹, 澤田 淳二}{Kentaro FURIHATA, Yoshiki TANIGUCHI, Junji SAWADA}
\endheader
\end{document}
```

Table 8 ヘッダの出力例



4.3 アブストラクトに関するコマンド

<code>\abstract</code>	<code>abstract</code> 環境は、レジュメの概要を記述するための環境です。概要は、英語で 4 行程度にまとめて記述してください。
------------------------	---

abstract の使用例

```
\begin{abstract}
This is a sample document which uses the
monthly lecture meeting at Mikilab. This is
a sample document which uses the monthly
lecture meeting at Mikilab. \end{abstract}
```

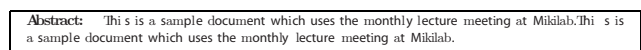


Fig. 15 abstract の出力例

Fig. 14 ソースファイル

4.4 参照に関するコマンド

Table 9 参照 .

Table 9 参照に関するコマンド

参照の種類	ソース	出力例
通常の参照	<code>\ref{(参照名)}</code>	1.1
図の参照	<code>\fgref{(参照名)}</code>	Fig.1.1
表の参照	<code>\tbref{(参照名)}</code>	Table.1.1
数式の参照	<code>\eqref{(参照名)}</code>	式 (1.1)

4.5 フォントおよびサイズに関するコマンド

Table 16 を参照 .

フォント名 ²	ソース	出力例
ボールド体	<code>\textbf{(文字列)}</code>	ボールド体 BoldFace
タイプライタ体	<code>\texttt{(文字列)}</code>	タイプライタ体 <code>TypeWriter</code>
イタリック体	<code>\textit{(文字列)}</code>	イタリック体 <i>Italic</i>
ゴシック体	<code>\textgt{(文字列)}</code>	ゴシック体

Fig. 16 フォントの変更

ソース	出力例
<code>{\small Sample}</code>	Sample
<code>{\large Sample}</code>	Sample
<code>{\Large Sample}</code>	Sample
<code>{\LARGE Sample}</code>	Sample
<code>{\huge Sample}</code>	Sample
<code>{\Huge Sample}</code>	Sample

Fig. 17 サイズの変更

4.6 脚注に関するコマンド

脚注を表示するには、`\footnote` 命令を用います .

ソース	
このページの下 <code>\footnote{これが脚注です}</code> にあるのが脚注です .	
出力	
このページの下 ^a にあるのが脚注です .	
<hr style="width: 20%; margin-left: 0;"/> ^a これが脚注です	

4.7 見出しに関するコマンド

<code>\chapter</code>	章の設定であり、部の次に大きいまとまりであります . しかし、 <code>jarticle</code> クラスと <code>tarticle</code> クラスには用意されていません .
<code>\section</code>	節の設定であり、3 番目に大きいまとまりであります .
<code>\subsection</code>	項 (小節) の設定であり、4 番目に大きいまとまりであります .
<code>\subsubsection</code>	目 (小々節) の設定であり、5 番目に大きいまとまりであります .

具体例：見出しの使用

```
\documentclass[a4paper,10pt]{jreport}
\usepackage{mikilab}
\pagestyle{empty}
\begin{document}
\chapter{これが章です。}
\section{これが節です。}
\subsection{これが小節です。}
\subsubsection{これが小々節です。}
\end{document}
```

Fig. 18 ソースファイル

第1章 これが章です。

1.1 これが節です。

1.1.1 これが小節です。

1.1.1.1 これが小々節です。

Fig. 19 Fig. 18 の実行結果

4.8 空白調節に関するコマンド

<code>\hspace{}</code>	命令の書かれた行と次行とのあいだを，引数の値だけ空ける。
<code>\vspace{}</code>	命令の書かれた箇所を，引数の値だけ空ける。

具体例：空白調節の使用

```
\section{空白調節なし}
かなり空いています。
```

Fig. 20 調節前のソースファイル

第1章 空白調節なし

かなり空いています。

Fig. 21 Fig. 20 の実行結果

```
\section{空白調節あり}
\vspace{-15mm}
かなり空いています。
```

Fig. 22 調節後のソースファイル

第1章 空白調節あり

狭くなりました。

Fig. 23 Fig. 22 の実行結果

4.9 環境に関するコマンド

documentclass 命令	「あらかじめ定義された設定」を選択し，文書の基本的な体裁を指定する命令であります。具体的には，文書の種類（文書クラス）とそのオプション（クラスオプション）を定義する働きをもっています。
document 環境	L ^A T _E X に「そこが組版し，出力すべき文章である」ことを伝えます。また，documentclass 命令と document 環境の間の領域のことを「プリアンプル」といいます。

L^AT_EX の基本原稿構造

```

\document[クラスオプション]{文書オプション}
プリアンプル
\begin{document}
原稿の本体
\end{document}

```

Fig. 24 基本構造

figure 環境	「図版を張り込む領域を確保する環境」です。figure 環境の中には、実際の図表を出力させるための\includegraphics 命令や、必要に応じて\caption 命令や\label 命令を指定します。後者の 2 つは必ず指定しなければならないわけではありませんが、文書全体をわかりやすくするために、極力指定しましょう。
\caption{}	図版の説明文を指定します。{} の中に文を入れるのですが、長い文章になると図版の下に記述されます。それを避けたいのならば、\caption 命令を実際の図版より前に記述してください。
\includegraphics[[]]{}	図版を貼り込むための命令です。[] にはオプション引数として、貼り込み後の画像サイズを、{} には引数として、貼り込みたい画像ファイル名 (eps) が必要となります。。
\label{}	図版の参照用の目印を付け加えます。

figure 環境 の基本原稿構造

```

\begin{figure}
    \caption{図版の説明文}
    実際の図版
    \label{ラベル}
\end{figure}

```

Fig. 25 figure 環境

table 環境	「表を張り込む領域を確保する環境」です。詳細は figure 環境と同じなので省略します。なお、二段組みの文書内で二段抜きの表を作成したい場合は、table*環境を用いてください。
----------	--

tabular 環境	同レジюме第 2 章に詳細があります。
------------	----------------------

verbatim 環境	\verb 命令によって、特殊文字を出力させることができました。しかし、プログラムリストのように複数行に渡る原稿をそのまま出力させるときは面倒です。このようなときに、verbatim 環境を使います。この環境内に書かれた文章は、そのまま出力されます。
-------------	---

itemize 環境	itemize 環境は、複数の項目を見出し付き箇条書きにしたいときに用います。itemize 環境の中では、各項目は\item 命令を使用して並べます。また、箇条書きは第 4 レベルまで入れ子状態にすることも可能です。
------------	---

```
\begin{itemize}
\item 第 1 レベルの項目 1
\item 第 1 レベルの項目 2
\begin{itemize }
\item 第 2 レベルの項目 1
\item 第 2 レベルの項目 2
\end{itemize}
\item 第 1 レベルの項目 3
\end{itemize}
```

Fig. 26 ソース

- 第 1 レベルの項目 1
- 第 1 レベルの項目 2
 - 第 2 レベルの項目 1
 - 第 2 レベルの項目 2
- 第 1 レベルの項目 3

Fig. 27 出力

enumerate 環境	番号などを付けた箇条書きのための環境です。itemize 環境同様、enumerate 環境も入れ子（第 4 レベルまで）にすることが可能です。
--------------	--