

TEX

マニュアル

目次

1	TEXの基礎	3	2.3	見出し記号付きの箇条書き (itemize)	12
1.1	TEXの誕生	3	2.3.1	見出しのマークを変更する	13
1.2	TEXで何ができるのか	3	2.4	番号付きの箇条書き (enumerate)	14
1.2.1	カーニング	3	2.4.1	番号の書式を変更する	14
1.2.2	合字	3	2.5	見出し語付きの箇条書き (description)	15
1.2.3	ハイフネーション	3	2.6	文書揃え	16
1.2.4	文字配置の制御	3	2.6.1	中央揃え	16
1.2.5	数式	3	2.6.2	左寄せ	16
1.3	視覚デザインと論理デザイン	3	2.6.3	右寄せ	16
1.3.1	視覚デザイン	3	2.7	そのまま出力 (verbatim)	16
1.3.2	論理デザイン	4	2.8	脚注	16
1.4	TEXでの約束ごと	4	2.9	番号の初期値を変更する	17
1.4.1	ファイル名	4	2.10	2段組で書いたとき、本文の最後を揃える	17
1.4.2	最低限のルール	4	2.11	文字の色を変更する	17
1.5	TEXの原稿	5	2.11.1	さらに多くの色を利用する	18
1.5.1	ソースファイルの中身	5	3	図	20
1.5.2	環境と命令	5	3.1	図の挿入	20
1.5.3	英文の空白	6	3.2	配置の指定	20
1.5.4	改行・段落	7	3.3	図のキャプション	21
1.5.5	改ページ	7	3.3.1	番号の前に付記される文字を変更する	21
1.6	\documentclass 命令	7	3.3.2	図 1: の部分を太字にする	21
1.6.1	文書クラス	7	3.3.3	キャプション番号に節番号をつける	21
1.6.2	クラスオプション	8	3.4	図を複数枚並べて出力	21
1.7	プリアンプル	9	3.4.1	2段抜き	21
1.7.1	標題	9	3.4.2	minipage 環境	21
1.7.2	ページ番号の出力	9	3.4.3	図を2枚横に並べて出力する	22
1.7.3	\usepackage 命令	9	3.4.4	図を横に2枚、縦に2枚並べて出力する	22
1.8	エラーメッセージ	10	3.4.5	図を3枚横に並べて出力する	23
1.9	TEXで扱う長さの単位	10	3.5	フロートのレイアウト	23
2	本文の編集	12	3.5.1	1ページ内のフロートの数を変える	23
2.1	フォント	12	3.5.2	本文とフロートの割合	24
2.2	文字の大きさ	12			

3.6	図と表を並べて出力し、それぞれにキャプションをつける	26	7	PDF 変換	40
4	表	30	7.1	PDF ファイルを作る	40
4.1	表の挿入	30	7.2	レターサイズで変換するときの注意 . . .	40
4.1.1	要素の配置	30	7.3	PDF ファイルを作成するときの注意 . .	40
4.1.2	要素に長い文章を入れる	31			
4.2	罫線	31			
4.2.1	縦罫線の引き方	31			
4.2.2	横罫線の引き方	31			
4.2.3	部分的に横罫線を引く	31			
4.2.4	横の二重線を引く	32			
4.3	セルを結合する	32			
4.3.1	横方向のセルの結合	32			
4.3.2	縦方向のセルの結合	33			
4.4	セルの背景に色をつける	33			
5	数式	34			
5.1	数式環境	34			
5.1.1	単純数式モード	34			
5.1.2	別行立て数式モード	34			
5.2	空白	34			
5.3	添え字、べき乗	34			
5.4	総和・総積	34			
5.5	分数	34			
5.6	積分	35			
5.7	複数行の数式を揃える	35			
5.8	長い数式を書く	36			
5.9	行列	36			
5.10	場合分け	37			
5.10.1	行列を使う場合	37			
5.10.2	cases 環境を使う場合	37			
5.11	本文中で数式を大きく表示する	37			
5.12	カッコを大きく出力する	37			
5.13	式番号の表示方法を変える	38			
6	EPS ファイル	39			
6.1	EPS ファイルの作成方法	39			
6.1.1	Illustrator を使う	39			
6.1.2	WMF2EPS を使う	39			

1 TeXの基礎

1.1 TeXの誕生

TeXはアメリカ・スタンフォード大学の数学者 Donald E. Knuth 博士によって開発された文書組版システムです。Knuth 博士は1968年から「The Art of Computer Programming」という本を書き始め、第2巻からは実際の活字を組み合わせて印刷用の版を作成する活版印刷ではなく、コンピュータによる組版によって版の作成が行われました。

ところが、当時のコンピュータによる組版結果は、活版印刷によるものと比べ数式などの仕上がりが悪く、Knuth 博士は大変失望したそうです。そのため、Knuth 博士はひとまず著書の出版を見合わせ、特に組版が難しいとされる技術関係の書籍を美しく組版できるようなソフトウェアを作ることにしました。そして、ほぼ4年の歳月を経てできあがったのが TeX という組版システムです。

1.2 TeXで何ができるのか

TeXは組版をするためのソフトウェアですから、文字を配置することにかけては大概のことはできます。その能力はワープロとは比較になりません。このマニュアル自体、TeXで作成しました。

1.2.1 カーニング

欧文を組版するとき、隣り合う文字の組み合わせによって字間を調節し、視覚上文字が等間隔に並んでいるように見せるための処理を「カーニング (kerning)」といいます。例えば、“To”、“Ye”、“AV” などのように2つの文字が微妙に重なっているものを指します。TeXはこの処理を自動的に行います。

1.2.2 合字

“ff” や “fi” のように組み合わせで英文字を使用する場合、“ff”、“fi” のように複数の文字を一体化した特殊な字形を使用します。このような字形を「合字 (ligature)」といい、TeXはこのような処理も自動で行ってくれます。

1.2.3 ハイフネーション

日本語にはハイフンはありませんが、コンピュータで英文を組版するときには必要になります。ワープロの中には、単語間の空白や文字送りを伸縮して、単語間を改行位置にする機能を持っていますが、これは結果として1行の文字数が少なくなったり、単語間の空白や文字送りが間延びすることも少なくありません。

1983年に Franklin Mark Liang がハイフネーション方法を提案しており、実際、ウェブスターの辞書の90%ほどを正確にハイフネーションすることができるそうです。TeXでは、この方式が実装されており、必要に応じて自動的にほぼ正確なハイフネーションが可能となっています。

1.2.4 文字配置の制御

TeXは文字を非常に細かい精度 (1スケールドポイント=約 5.4×10^{-6} mm) で配置することができます。文字を上や下に微妙にずらすという文字配置も、簡単な命令だけで制御することができます。

上の例：文字を上上げる

```
\raisebox{0.75ex}[0pt][0pt]{上}
```

1.2.5 数式

例えば、Word などには数式エディタが添付されていますが、出力のバランスなどの美しさはTeXの比ではありません。TeXは数式を綺麗に出力することを主に開発されているため、数式を美しくかつ自由自在に出力することができます。

$$f(x) = \int_0^{\infty} g(t) dt + \frac{1}{T} \sum_{k=0}^{T-1} \left(x_k^2 - \frac{h^{(k)}(x)}{k!} \right)$$
$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

1.3 視覚デザインと論理デザイン

1.3.1 視覚デザイン

視覚デザインとは、最近 “What You See Is What You Get (見たままに出力できる)” (略して “WYSIWYG”) と

呼ばれるデザイン手段です。この方法では、ディスプレイ上で編集しているものとまったく同じものが出力されます。希望通りに文字を配置し、自分で整形することができます。しかし、文書の規模が一定以上になるとこのようなデザインの方法では効率が悪くなります。(例えば、「見出しの文字の大きさは何ポイントだっけ?」とか「図や式に通し番号をつけるが、前の番号はいくつだっけ?」…)

また、図や式、節を追加・削除したとき「図 2 に示すように~」とか「式 (8) の右辺に代入すると~」といった参照を含めて、通し番号を振り直さなければなりません。

WYSIWYG システムは、画面上で見ているとおりの出力を得られるという点では非常に便利ですが、その反面 “What You See Is All You’ve Got(見たとおりの出力しかせず、ミスの有無には関知しない)” システムだと揶揄(やゆ)されることもやります。

1.3.2 論理デザイン

論理デザインは、文章の持つ論理構成にのみ注目して文書をデザインする方法で、一般で “マークアップ方式” などと呼ばれています。マークアップ方式を採用している代表的なものに HTML(Hyper Text Markup Language) があります。Web ページは、原稿である HTML 文書に記述されている HTML の命令に従って、Web ブラウザが自動的にページの出力を行います。T_EX の場合、文章の論理構成に応じて原稿中に埋め込まれた T_EX 命令に従って、T_EX が自動的に体裁を整えて文書を出します。

これが論理デザインの最大の特徴といえます。論理デザインは中央揃えであるとか、箇条書きであるといった文章の論理的な構成をあらかじめ何らかの手段でマークアップしておくことで、体裁に依存しない文書を作成することができます。

T_EX の場合、文章の理論的な構成は「環境」と呼ばれる命令を通じて T_EX に伝えられます。また、環境のほかにも文章の構成要素(章や節)を指定する命令や、参照用の通し番号を付けたり、その通し番号を参照する命令なども用意されています。

1.4 T_EX での約束ごと

1.4.1 ファイル名

T_EX では、原稿ファイルの拡張子を

`*.tex`

にするという約束があります。

ファイル名に特別な決まりがありませんが、“~”(チルダ)という文字を利用することはできません。また、原稿ファイル名に半角の空白文字を含めることはできません。これは、T_EX が半角の空白文字を区切りの文字として利用しているからです。全角文字については、原則的に制限がありませんが、文字コードの都合上からエラーが発生することもあります。

1.4.2 最低限のルール

(1) T_EX の命令は原則として “~~≠~~” で始まり、空白文字で終わる

T_EX の組版命令は、ほとんどが “\” で始まります。命令に続く文字が “全角の空白や句読点”, “全角および半角のカッコや記号”, “半角の数字” であれば半角の空白は必要ありません。

例えば、

`αx`

と書く場合、

`\alpha_x`

とします。

`\alphax`

と続けて書いてしまうと T_EX は `alphax` という命令と解釈してしまい、そのような命令は定義されていない “! Undefined Control Sequence” というエラーメッセージを出力します。

(2) 半角の空白はいくつ続いても 1 つみなされる

“\TeX_” と書いた場合も “\TeX_” と書いた場合も、出力上は同じ結果となります。また、改行直後(行頭)の半角空白は無視されます。

(3) 全角文字の直後の改行 無視, 半角文字の直後の改行 空白。また、2 つ以上続いた改行 段落の切れ目(出力結果上の改行)

```

\documentclass{jarticle}

\begin{document}
本日は
晴天なり。
It's fine
today.

でも、明日は雨でしょう。
\end{document}

```

と入力すると、

```

本日は晴天なり。It's fine today.
でも、明日は雨でしょう。

```

と出力されます。

(4) 半角記号の中には、そのまま入力しても出力できない文字がある

表 1.1 に示す半角記号は、そのまま入力しても $\text{T}_{\text{E}}\text{X}$ では出力できません。それぞれに対応する命令を用いてください。また、全角の場合はそのまま出力されます。

(5) 半角カタカナは使用できない

現在の $\text{T}_{\text{E}}\text{X}$ では、半角カタカナを入力してもエラーにはなりません、正しく出力されません。また、Windows の機種依存文字である“丸付き数字”や“全角ローマ数字”は DVIOUT には出力できますが、PDF 変換後は出力されません。

(6) %以降は改行コードも含めてコメント

行中の%記号以降は、改行コードも含めてコメントとして扱われます。実際の実稿では改行しているのに、改行してないのと同様に処理したい場合などに%記号を利用します。

(7) $\text{T}_{\text{E}}\text{X}$ 原稿中に必ず記述しなければならない命令がある

$\text{T}_{\text{E}}\text{X}$ の原稿中には、

- `\documentclass`
- `\begin{document} ... \end{document}`

という命令が記述されていなければなりません。

1.5 $\text{T}_{\text{E}}\text{X}$ の原稿

1.5.1 ソースファイルの中身

$\text{T}_{\text{E}}\text{X}$ の原稿ファイルについて説明します。

```

\documentclass{jarticle}

    プリアンプル

\begin{document}

    原稿

\end{document}

```

1 行目の `\documentclass{jarticle}` は、これから記述する文章の種類や書式を指定しています。

実際の文章は `\begin{document}` から `\end{document}` に中に記述します。`\end{document}` より後ろに原稿を書いても処理されません。

`\begin{document}` よりも前の部分はプリアンブルと呼ばれる領域です。この部分には原稿を処理する前に $\text{T}_{\text{E}}\text{X}$ に対して与える指示を記述します。そのため、この部分に原稿を書くとエラーが発生します。

1.5.2 環境と命令

$\text{T}_{\text{E}}\text{X}$ は論理デザインが可能のため「中央揃え」や「箇条書き」といった文章の構成はあらかじめ用意されている「環境」と呼ばれるマクロ命令を通じて $\text{T}_{\text{E}}\text{X}$ に伝えられます。

```

\begin{環境名}

\end{環境名}

```

のようになつた命令を環境といいます。環境の内側は本文とは別の世界となり、色々な設定が環境の外側と異なります。例えば、`itemize` 環境を使用すると箇条書きのスタイルになります。

```

\begin{itemize}
\item 箇条書きその1
\item 箇条書きその2
\end{itemize}

```

表 1.1: T_EX で単純に出力できない記号

記号	原稿中の働き	命令	出力
#	命令や環境の引数	\#	#
\$	数式モードへの移行	\\$	\$
%	以降をコメント	\%	%
&	表のセルの区切り	\&	&
_	下付の添え字	_	-
{	グループ化の開始	\{	{
}	グループ化の終了	\}	}
<	段落モードで“<”を出力	\textless	<
>	段落モードで“>”を出力	\textgreater	>
\	命令のためのエスケープ文字	\backslash	\
—	表の縦の罫線	\textbar	
^	上付きの添え字	\textasciicircum	^
~	改行を抑制して単語間の空白を空ける	\textasciitilde	~

上の出力例：

- 箇条書きその 1
- 箇条書きその 2

命令は

`\命令 [オプション引数]{引数}`

の形で記述します。[と] で囲まれたオプション引数は省略可能です。数学記号を出力する、見出しを付けるなど様々な命令があります。

1.5.3 英文の空白

T_EX は英文の単語や文の区切りで自動的に適量の空白を挿入します。単語間の空白より文の区切りの空白だけ広く確保されます。

T_EX は、., ?, !, ; の後ろには常に空白量を余分にとります。また, ;, , の後ろは単語間の空白よりも広めによることがあります。

そのため、省略を表す“etc.”や、“Dr. Knuth”を入力したときは“.”をそれぞれ文の終わりだと判断してしまいます。そのため、省略を表すピリオドの後ろには `_` 命令や `~` を書いて単語間の空白を挿入します。

```
I read Dr. Knuth's book.
I read Dr. \Knuth's book.
I read Dr. ~Knuth's book.
```

と入力すると、それぞれ

```
I read Dr. Knuth's book.
I read Dr. Knuth's book.
I read Dr. Knuth's book.
```

と出力します。

ところで、1.4.2 の (1) で述べた命令のあとには、原則的に命令の終わりを示すための半角スペースを挿入しなければなりません。従って、T_EX の命令の直後に半角スペースを空けても、単語間の自然な空白は飽きません。この場合、T_EX に半角スペース以外の方法で命令の終わりを示すか、空白を空ける命令を使用します。

- (1) `\TeX for Windows`
- (2) `\TeX\ for Windows`
- (3) `\TeX~ for Windows`
- (4) `\TeX\@ for Windows`
- (5) `\TeX{ } for Windows`
- (6) `{\TeX} for Windows`

と入力すると、それぞれ

- (1) T_EXfor Windows
- (2) T_EX for Windows
- (3) T_EX for Windows
- (4) T_EX for Windows
- (5) T_EX for Windows
- (6) T_EX for Windows

と出力されます。それぞれの違いは、

- (1) 命令直後の半角空白は命令の終端を表すため空白は無視
- (2) 空白を空ける命令`_`を使用
- (3) 空白を空ける命令`~`を使用
- (4) 「何も出力しない」という`\@`命令を使用
- (5) “{}”と記述した場合、何も出力されませんが、T_EXは何らかの区切りであると認識します。
- (6) 命令を“{}”で囲むとT_EXはその部分で命令が終わっていると認識します

1.5.4 改行・段落

T_EXは、連続する改行記号を段落の区切りとして判断します。つまり、何も記述されていない改行記号だけの行があれば、それを段落の区切りとして見なします。段落の始まりでは、英文の場合には適当な分量だけ、和文の場合は全角1文字分の字下げを自動的に挿入します。

なお、段落を変えるのではなく、単に改行したい場合は改行したい場所で`\newline`を用いるか`\\`命令を使用します。(この2つの命令は同じ働きをします。)

実際には、以下のように入力すると

桃太郎さん、桃太郎さん、
お腰につけたきび団子、
一つわたしに下さいな。

やりませう、やりませう、
これから鬼の征伐に、
ついて行くならやりませう。\\
行きませう、行きませう、
あなたについてどこまでも、
家来になって行きませう。

次のように出力されます。

桃太郎さん、桃太郎さん、お腰につけたきび団子、一つわたしに下さいな。

やりませう、やりませう、これから鬼の征伐に、ついて行くならやりませう。

行きませう、行きませう、あなたについてどこまでも、家来になって行きませう。

1.5.5 改ページ

≠`newpage`

強制的に改ページしたい場合、その場所で`\newpage`命令を使用します。2段組みの場合で、現在位置が左側の段の場合は右側の段に移動します。

≠`clearpage`

強制的に改ページしたい場合、その場所で`\clearpage`命令を使用します。`\newpage`と違い、`\clearpage`の使用時に配置が決定されていない図や表があれば、それらをすべて出力してから改ページします。2段組みの場合でも、常に新しいページから始まります。

1.6 ≠`documentclass` 命令

T_EXの原稿は必ず次のような構造から成り立っています。

```
\documentclass[クラスオプション]{文書クラス}
  
    プリアンプル
  
\begin{document}
  
    原稿
  
\end{document}
```

T_EXでは、まず1行目で文書の種類(文書クラス)とそのオプション(クラスオプション)を宣言します。

1.6.1 文書クラス

文書の種類を指定します。この文書クラスが指定されると“クラスファイル”という拡張子が`.cls`というファイルを読み込みます。この中には、文書の体裁のほか、文書に依存する構成要素や論理構造などの情報が定

義されています。

TeX では、文書クラスとして以下のものが指定できます。

`article`, `jarticle`, `tarticle`

“article” とは、「記事」や「論説」といった意味で、比較的短い文書のことです。学会の予稿集やジャーナルはこのクラスで執筆します。

`article` クラスは欧文、`jarticle` クラスは横書きの和文、`tarticle` クラスは縦書きの和文に対応します。

`report`, `jreport`, `treport`

“report” は、比較的長い「報告書」などを意味します。

`report` クラスは欧文、`jreport` クラスは横書きの和文、`treport` クラスは縦書きの和文に対応します。

節 (`\section`) の上のクラスである章 (`\chapter`) が使用できます。

`book`, `jbook`, `tbook`

“book” は、書籍 (本) のことをいいます。

`book` クラスは欧文、`jbook` クラスは横書きの和文、`tbook` クラスは縦書きの和文に対応します。

見出しは原則的に章 (`\chapter`) から始まります。`report` クラスとは異なり、奇数ページと偶数ページでは、左右の余白や柱の文字列等の点でデザインが異なります。

1.6.2 クラスオプション

文書の体裁の細かな設定は「クラスオプション」として設定します。ここで指定する事項には、例えば用紙の大きさや本文の文字の大きさなどがあります。特に指定しないと、デフォルトの値が設定されます。

複数のオプションを指定したいときは“,” で区切ります。

```
\documentclass[b4paper,11pt]{jarticle}
```

と設定すると、用紙サイズが B4、本文の文字サイズが 11pt に設定されます。

本文の文字の大きさ

最も基本となる文字の大きさ (`\normalsize`) を指定します。(表 1.2)

表 1.2: 本文の文字サイズ

文字サイズ	オプション	備考
10 ポイント	10pt	デフォルト
11 ポイント	11pt	
12 ポイント	12pt	

用紙のサイズ

出力する用紙の大きさを指定します。(表 1.3)

表 1.3: 用紙サイズ

用紙サイズ	オプション	備考
A4	a4paper	pLaTeX のデフォルト
A5	a5paper	
B4	b4paper	
B5	b5paper	

用紙の方向

出力する用紙の向きを指定します。(表 1.4)

表 1.4: 用紙の方向

用紙の方向	オプション	備考
縦置き	指定しない	デフォルト
横置き	landscape	

2 段組み

ページ内の段組みを指定します。(表 1.5)

表 1.5: 2 段組み

段組み	オプション	備考
1 段組み	onecolumn	デフォルト
2 段組み	twocolumn	

両面印刷

これらの指定はプリンタとは関係ありません。表と裏面で余白の量を変えたいときなどに使用します。(表 1.6)

数式の配置

別行立ての数式 (`equation` 環境など) は、デフォルトでは行の中央に配置されています。(表 1.7)

表 1.6: 両面印刷

印刷面	オプション	備考
片面印刷	oneside	デフォルト
両面印刷	twoside	book 系クラスのデフォルト

表 1.7: 数式の配置

式の配置	オプション	備考
中央揃え	指定しない	デフォルト
左寄せ	fleqn	字下げ幅は $\backslash\mathindent$ に格納

式番号の位置

数式番号の位置を指定します。(表 1.8)

表 1.8: 式番号の位置

数式番号の位置	オプション	備考
用紙の右端	指定しない	デフォルト
数式の左側	leqno	

1.7 プリアンブル

$\backslash\documentclass$ 命令から $\backslash\begin\{document\}$ までの間を「プリアンブル」と呼びます。この部分では、

- 原稿全体で使用される変数やマクロの定義
- スタイルファイルの読み込み
- 標題を出力するための命令
- 文章中で使用する特殊な命令

などを設定します。

1.7.1 標題

$\text{T}_\text{E}_\text{X}$ の標準的な文書クラスには、標題を出力するための $\backslash\maketitle$ 命令が用意されています。この命令を実行する前に、タイトル ($\backslash\title$)、著者名 ($\backslash\author$)、日付 ($\backslash\date$) を定義しておきます。

```
\documentclass{jarticle}
\title{\TeX マニュアル}
\author{山田 太郎}
\date{2006 年 1 月 1 日}
\begin{document}

\maketitle

\end{document}
```

タイトルが長い場合やサブタイトルを付けたい場合は改行命令 $\backslash\backslash$ を書きます。

著者が複数いる場合は $\backslash\and$ 命令で区切ります。

1.7.2 ページ番号の出力

ページ番号の出力する / しないは

```
\pagestyle{引数}
```

命令を使用します。

表 1.9: ページ番号の出力

引数	意味
plain	各ページの下部に出力します。
empty	ページ番号を出力しません。
headings	ページ番号のほかに、ページ上部に柱を出力します。

ページ番号はの初期値は 1 ですが、プリアンブルで

```
\setcounter{page}{0}
```

と設定すると 0 ページから始まります。

また、 $\backslash\pagenumbering\{引数\}$ 命令で表 1.10 の引数を指定することで数字の書体を変更することができます。

1.7.3 $\backslash\usepackage$ 命令

$\text{T}_\text{E}_\text{X}$ 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 (拡張子 $.sty$) を読み込むことで、命令を変更したり追加することができます。予稿集などを執筆するときは、学会が提供しているスタイルファイルを読み込むことで文書スタイルなどが統一されたフォーマットにすることができます。

表 1.10: 数字の書体

引数	書体
arabic	算用数字 (1, 2, 3, ...)
roman	ローマ数字小文字 (i, ii, iii, ...)
Roman	ローマ数字大文字 (I, II, III, ...)
alph	英小文字 (a, b, c, ...)
Alph	英大文字 (A, B, C, ...)

“foo.sty” というスタイルファイルを読み込むときは、該当するファイルをソースファイルと同じフォルダに格納し、プリアンプル部で

```
\usepackage{foo}
```

と記述します。

1.8 エラーメッセージ

TeX はプログラミングのようにコンパイルして結果 (文書) を出力するため、比較的エラーが起こりやすく、出てくるエラーも限られています。エラーメッセージをよく読んで、原稿の修正をします。

! Undefined control sequence.

入力した命令が定義されていません。また、命令のスペルを間違えたり、命令と本文を続けて記述すると発生します。

! Missing \$ inserted.

数式モードでしか使えない命令 (ギリシャ文字や演算子) を数式モード以外で使用しています。多くの場合 \$ を書き忘れていました。

! Something's wrong—perhaps a missing \item.

箇条書き環境 (itemize や enumerate 環境) で \item 命令を書き忘れていました。

! Double superscript.

数式中で連続して上付き文字 (例えば, a^{i^j}) を指定しています。 a^{ij} としたい場合は $a^{\{i^j\}}$ と記述します。

! Double subscript.

数式中で連続して下付き文字 (例えば, a_{x_y}) を指定しています。 a_{xy} としたい場合は $a_{\{x_y\}}$ と記述します。

! \begin{...} ended by \end{...}.

```
\begin{center}
...
\end{flushleft}
```

のように \begin と \end で環境が違うときに発生します。

! Too many unprocessed floats.

TeX が適当な改ページ位置を発見できなかった場合のエラーです。具体的には, figure 環境や table 環境で 1 ページに図表を大量に組版しようとしたときに生じます。

figure 環境や table 環境で図版の個数が多すぎる場合は、図表のまえで適当に強制改ページを挿入したり、いくつかの図表を後ろのページに送ったりするとエラーが解消されます。

! Extra }, or forgotten \$.

カッコ類の対応が合っていません。{, \[, \[, \$ を記入し忘れたか, },],), \$ を多く記入していることが原因です。

! Missing { または } inserted.

予想外のところにカッコが入力されているので TeX が混乱しています。どこかでカッコを記述し忘れた可能性が高いです。

たいていの場合、このエラーが出た数行前、あるいは数行後にエラー発生の原因があります。

1.9 TeX で扱う長さの単位

TeX で長さを扱うための単位として、表 1.11 のものが用意されています。

また、TeX には表 1.12 で示すような相対的な単位があり、フォントの何倍という形で指定できます。そのため、1em が常に同じ大きさであるとは限りません。

マクロで定義するときに絶対量で指定すると、スタイルによってはバランスが悪くなることがあるため、相対的な単位で指定の方が便利です。

表 1.11: T_EX で扱う長さの単位

単位	読み	他の単位との換算
pt	ポイント	1[pt] = 65536[sp] = 0.3514[mm]
sp	スケールドポイント	1[sp] = 0.54 × 10 ⁻⁵ [mm]
in	インチ	1[in] = 72[bp] = 72.27[pt] = 25.4[mm]
bp	ビッグポイント	1[bp] = 1.004[pt] = 0.3527[mm]
cm	センチメートル	1[cm] = 10[mm] = 28.34[pt]
mm	ミリメートル	1[mm] = 0.1[cm] = 2.83[pt]
pc	パイカ	1[pc] = 12[pt] = 4.2[mm]
cc	シセロ	1[cc] = 12[dd] = 12.835[pt] = 4.511[mm]
dd	ディドポイント	1[dd] = 0.376[mm] (1157[dd] = 1238[pt])
Q	級	1[Q] = 1[H] = 0.25[mm]
H	歯	1[Q] = 1[H] = 0.25[mm]

表 1.12: 相対的な単位

単位	意味
em	現在の欧文フォントにおける M の幅
ex	現在の欧文フォントにおける x の高さ
zw	現在の和文フォント (全角文字) の幅
zh	現在の和文フォント (全角文字) の高さ

2 本文の編集

2.1 フォント

出力可能な書体を表 2.1 に示します。フォントには半角英文字，全角文字にのみしか作用しないものがあるので注意が必要です。特に， $\text{T}_{\text{E}}\text{X}$ は全角文字のイタリック体は出力できません。

書体の変更は

```
\textrm{変更したい文字列}
```

ように記述します。

2.2 文字の大きさ

$\text{T}_{\text{E}}\text{X}$ で使用できる文字サイズを表 2.2 に示します。

文字サイズは基本となる本文の文字サイズ ($\backslash\text{normalsize}$) を基準としているため，クラスオプションで 10pt, 11pt, 12pt のどれを選択したかで実際のサイズが異なります。

2.3 見出し記号付きの箇条書き (itemize)

itemize 環境は，見出し付きの箇条書きをするための環境です。基本的には以下のように記述します。

```
\begin{itemize}
\item 本日は晴天なり。
\item 昨日は曇りでした。
\item 明日は雨でしょう。
\end{itemize}
```

出力は，次のようになります。

- 本日は晴天なり。
- 昨日は曇りでした。
- 明日は雨でしょう。

$\backslash\text{item}$ 命令が箇条書きの記号になります。

また，以下のように第 4 レベルまで入れ子にすることができます。

```
\begin{itemize}
\item 第 1 レベル その 1
\item 第 1 レベル その 2
\begin{itemize}
\item 第 2 レベル その 1
\begin{itemize}
\item 第 3 レベル その 1
\item 第 3 レベル その 2
\begin{itemize}
\item 第 4 レベル その 1
\end{itemize}
\end{itemize}
\item 第 3 レベル その 3
\end{itemize}
\item 第 2 レベル その 2
\end{itemize}
\item 第 1 レベル その 3
\end{itemize}
```

出力結果は次のようになります。

表 2.1: 使用できる書体と命令

書体名	命令	出力例
ローマン	<code>\textrm</code>	123abcABC あいうアイウ亜以宇
サンセリ	<code>\textsf</code>	123abcABC あいう亜以宇
タイプライター	<code>\texttt</code>	123abcABC あいう亜以宇
ミディアム	<code>\textmd</code>	123abcABC あいう亜以宇
イタリック	<code>\textit</code>	<i>123abcABC</i> あいう亜以宇
スラント	<code>\textsl</code>	<i>123abcABC</i> あいう亜以宇
スモールキャピタル	<code>\textsc</code>	123ABCABC あいう亜以宇
ボールド	<code>\textbf</code>	123abcABC あいう亜以宇
明朝体	<code>\textmc</code>	123abcABC あいう亜以宇
ゴシック体	<code>\textgt</code>	123abcABC あいう亜以宇

表 2.2: 文字サイズと命令

出力例	命令	10pt	11pt	12pt
とても小さい	<code>\tiny</code>	5pt	6pt	6pt
かなり小さい	<code>\scriptsize</code>	7pt	8pt	8pt
小さい	<code>\footnotesize</code>	8pt	9pt	10pt
やや小さい	<code>\small</code>	9pt	10pt	11pt
普通	<code>\normalsize</code>	10pt	11pt	12pt
やや大きい	<code>\large</code>	12pt	12pt	14pt
大きい	<code>\Large</code>	14pt	14pt	17pt
かなり大きい	<code>\LARGE</code>	17pt	17pt	20pt
とても大きい	<code>\huge</code>	20pt	20pt	25pt
特大	<code>\Huge</code>	25pt	25pt	25pt

- 第1レベル その1
- 第1レベル その2
 - 第2レベル その1
 - * 第3レベル その1
 - * 第3レベル その2
 - ・ 第4レベル その1
 - * 第3レベル その3
 - 第2レベル その2
- 第1レベル その3

2.3.1 見出しのマークを変更する

`itemize` 環境の各レベルの見出しとなる記号はクラスファイルの中で表 2.3 のように定義されています。

`itemize` 環境を使用する前にこれらの命令を再定義することで、見出しとなる記号を変更することができます。レベル1の記号を“ ”に変更するには

```
\renewcommand{\labelitemi}{ }
```

とします。すると、以下のように変更されます。

第1レベルのマークが変更されました。

また、一時的に変更する場合は`\item`命令にオプション

表 2.3: itemize 環境の見出し記号

レベル	命令	デフォルト
1	<code>\labelitemi</code>	<code>•(\textbullet)</code>
2	<code>\labelitemii</code>	<code>–(\bfseries\textendash)</code>
3	<code>\labelitemiii</code>	<code>*(\textasteriskcentered)</code>
4	<code>\labelitemiv</code>	<code>·(\textperiodcentered)</code>

引数 [] に指定します。

```
\begin{itemize}
\item 左は通常のマーク
\item[ ] 一時的にマークを変更しました。
\end{itemize}
```

と書くと、以下の用に出力されます。

- 左は通常のマーク
- 一時的にマークを変更しました。

2.4 番号付きの箇条書き (enumerate)

`enumerate` 環境は、番号付きの箇条書きをするための環境です。基本的には以下のように記述します。

```
\begin{enumerate}
\item うどん
\item そば
\item そうめん
\end{enumerate}
```

出力は、次のようになります。

1. うどん
2. そば
3. そうめん

`\item` 命令が箇条書きの記号になります。

また、以下のように第 4 レベルまで入れ子にすることができます。

```
\begin{enumerate}
\item 第 1 レベル その 1
\item 第 1 レベル その 2
\begin{enumerate}
\item 第 2 レベル その 1
\begin{enumerate}
\item 第 3 レベル その 1
\item 第 3 レベル その 2
\begin{enumerate}
\item 第 4 レベル その 1
\end{enumerate}
\end{enumerate}
\item 第 3 レベル その 3
\end{enumerate}
\end{enumerate}
\item 第 2 レベル その 2
\end{enumerate}
\item 第 1 レベル その 3
\end{enumerate}
```

出力結果は次のようになります。

1. 第 1 レベル その 1
2. 第 1 レベル その 2
 - (a) 第 2 レベル その 1
 - i. 第 3 レベル その 1
 - ii. 第 3 レベル その 2
 - A. 第 4 レベル その 1
 - iii. 第 3 レベル その 3
 - (b) 第 2 レベル その 2
3. 第 1 レベル その 3

2.4.1 番号の書式を変更する

`enumerate` 環境の見出しとなる番号の文字 (数字, ローマ数字, ...) と書式 (カッコでくる, 太字にす

表 2.4: enumerate 環境の第 1 レベルの命令

命令	意味
<code>enumi</code>	番号が格納されているカウンタ
<code>\theenumi</code>	カウンタ <code>enumi</code> を出力する命令
<code>\labelenumi</code>	出力する番号の書式が定義されている

る, ...) は変更することができます。変更するときは、以下の命令が使用されるのでそれらを再定義してやりま
す。表 2.4 では第 1 レベルのものを示しています。第 2~
4 レベルはそれぞれ `enumi` の部分が `enumii`, `enumiii`,
`enumiv` になります。

番号の文字を変更する場合は、

```
\renewcommand{\theenumi}{\Roman{enumi}}
```

のように、カウンタを出力する命令を再定義します。上
の例では、第 1 レベルの番号が大文字のローマ数字に
変更されます。書体は表 2.5 のものが使用できます。

表 2.5: 数字の書体

引数	書体
<code>\arabic</code>	算用数字 (1, 2, 3, ...)
<code>\roman</code>	ローマ数字小文字 (i, ii, iii, ...)
<code>\Roman</code>	ローマ数字大文字 (I, II, III, ...)
<code>\alph</code>	英小文字 (a, b, c, ...)
<code>\Alph</code>	英大文字 (A, B, C, ...)

番号の書体を変更する場合は、

```
\renewcommand{\labelenumi}{(\theenumi)}
```

のようにして番号の書式の再定義をします。上の例で
は、第 1 レベルの番号を半角の丸カッコでくくっていま
す。この書式の変更は

- 太字にする
- 斜体にする
- カッコでくくる
- 文字を付け加える

といった操作などができます。

アルゴリズムを記述したいときなどに、“Step 1:” と
したい場合は、

```
\renewcommand{\theenumi}{\arabic{enumi}}
\renewcommand{\labelenumi}
{\textit{Step \theenumi}:}
```

とします。(2 つ目の `\renewcommand` は横スペースの関
係で改行しましたが、実際はつなげて書きます。) する
と、出力は以下の用になります。

```
Step 1: 初期設定
Step 2: 何か処理をして
Step 3: 出力
```

2.5 見出し語付きの箇条書き (description)

`description` 環境は、見出し語を挙げて箇条書きに
する環境です。

`description` 環境は、次のように使用します。

```
\begin{description}
\item[日本語] Japanese
\item[英語] English
\item[フランス語] French
\end{description}
```

すると、以下のように出力されます。

```
日本語 Japanese
英語 English
フランス語 French
```

各項目は `\item` 命令で指定し、見出し語となる文字列を
オプション引数にします。

2.6 文書揃え

2.6.1 中央揃え

`center` 環境は、環境内に記述した原稿を中央揃えで出力します。

```
\begin{center}
この文章を中央に揃えます。
\end{center}
```

とすると、以下のように出力されます。

この文章を中央に揃えます。

2.6.2 左寄せ

`flushleft` 環境は、環境内に記述した原稿を左揃えで出力します。

```
\begin{center}
center 環境の中で
\begin{flushleft}
この部分だけ左に揃えます。
\end{flushleft}
ここは中央揃えです。
\end{center}
```

とすると、以下のように出力されます。

center 環境の中で
この部分だけ左に揃えます。
ここは中央揃えです。

中央揃えする `center` 環境の中でありながら、`flushleft` 環境の中の要素は左揃えで出力されます。

2.6.3 右寄せ

`flushright` 環境は、環境内に記述した原稿を右揃えで出力します。

```
\begin{flushright}
この文章を右に揃えます。\\
改行しても揃ってます。
\end{flushright}
```

とすると、以下のように出力されます。

この文章を右に揃えます。
改行しても揃ってます。

2.7 そのまま出力 (verbatim)

`verbatim` 環境は、環境内に書いた原稿をそのまま出力します。通常、原稿中にそのまま記述できない文字 ($\$$ や $\%$ や \TeX の命令) も書いた通りに出力します。例えば、

```
\begin{center}
この文章を中央に揃えます。
\end{center}
```

は

```
\begin{verbatim}
\begin{center}
この文章を中央に揃えます。
\end{center}
\end{verbatim}
```

と記述しました。(通常は `\begin{center}` と記述すると \TeX はこれを命令とみなし出力しません。)

2.8 脚注

このページの下¹にあるような脚注を出力するには、`footnote` 命令を用いて

```
このページの下\footnote{これが脚注だ}に...
```

のように記述します。

`\footnote[] {脚注のコメント}` の大カッコ `[]` 内に数字を挿入すると脚注の番号が変わります⁵。

また、図 2.1 の記述をプリアンプル部に記述し、`\footnote[0] {脚注のコメント}` と指定すると、脚注の番号を付けずに出力することができます。ここで、上の記述をスタイルファイル内で記述する場合は `\makeatletter`、`\makeatother` を削除します。(`\$` `\dag` 命令などで別途脚注マーク “†” を数式環境の

¹これが脚注だ

⁵さっきの続きなので、通常は 2 番ですが、[5] にしてみました。

0 番を指定したときの脚注のコメント (本文中にも脚注番号は出力


```

\def\thefootnote{\ifnum\c@footnote>\z@\textasteriskcentered\@arabic\c@footnote\fi}
\makeatletter
\renewcommand{\footnoterule}{%
  \kern-3\p@
  \hrule width 0.4\columnwidth
  \kern 2.6\p@}
\def\thefootnote{\ifnum\c@footnote>\z@\@arabic\c@footnote\fi}
\makeatother

```

図 2.1: 脚注の番号を消すためのマクロ

上付き文字でつける必要があります。)† ここではマクロを付けたい場所で $\{\}\^{\dag}$ と書きました。

2.9 番号の初期値を変更する

通常、1 から始まる節や数式の番号を変更します。基本的には、プリアンブル部で

```
\setcounter{カウンタ}{値}
```

とします。カウンタは表示する直前にインクリメントするため、一つ少ない値を設定します。

例えば、3 節から始める場合は

```
\setcounter{section}{2}
```

2.5 節から始める場合

```

\setcounter{section}{2}
\setcounter{subsection}{4}

```

とします。

TeX では表 2.6 に示すカウンタが用意されています。

2.10 2 段組で書いたとき、本文の最後を揃える

2 段組 (twocolumn) で書いた文章の最後のページの本文の高さを揃えます。balance パッケージが別途必要になりますので、balance.sty を原稿ファイルと同じフォルダに格納します。

プリアンブル部で

されません)

```
\usepackage{balance}
```

と記述し、本文の頭で

```
\balance
```

と書きます。

2.11 文字の色を変更する

通常、TeX は「白地に黒字」ですが、color パッケージを使用することで様々な色を使用することができます。color パッケージの本体は、color.sty ファイルです。color.sty ファイルを他のパッケージと同様にプリアンブル部で

```
\usepackage{color}
```

のように読み込みます。

文字の色を変更するには `\color`、`\textcolor` 命令を使用します。

```
\color{色名}
```

とすると、それ以降の文字の色がすべて変更され、

```
\textcolor{色名}{文字列}
```

とすると、`{ }` で囲った文字列の色を変更します。ここで、色名として表 2.7 のものが指定できます。

表 2.6: T_EX で用意されているカウンタ

カウンタ名	管理する番号	利用している命令
part	部	\part
chapter	章	\chapter
section	節	\section
subsection	項	\subsection
subsubsection	目	\subsubsection
paragraph	段落	\paragraph
subparagraph	小段落	\subparagraph
page	ページ	
equation	数式	equation 環境
figure	図	figure 環境
table	表	table 環境
footnote	脚注	\footnote
enumi	番号付き箇条書き (レベル 1)	enumerate 環境
enumii	番号付き箇条書き (レベル 2)	enumerate 環境
enumiii	番号付き箇条書き (レベル 3)	enumerate 環境
enumiv	番号付き箇条書き (レベル 4)	enumerate 環境
mpfootnote	ミニページ環境の脚注	minipage 環境の\footnote

表 2.7: デフォルトで使用できる色名

色名	色
black	黒
white	白
red	赤
green	緑
blue	青
cyan	シアン
magenta	マゼンタ
yellow	黄色

```
\color [named] {色名}
\textcolor [named] {色名} {文字列}
```

色名として、表 2.8 に示すものを指定します。

2.11.1 さらに多くの色を利用する

DVIPS では、合計 68 色の色が最初から定義されています。これらの色は DVIPS と互換性のある DVIOUT でも利用することができます。\\color, \\textcolor 命令で、以下のように named オプションを設定します。

表 2.8: DVIPS で定義されている色名

GreenYellow	Yellow	Goldenrod	Dandelion	Apricot
Peach	Melon	YellowOrange	Orange	BurntOrange
Bittersweet	RedOrange	Mahogany	Maroon	BrickRed
Red	OrangeRed	RubineRed	WildStrawberry	Salmon
CarnationPink	Magenta	VioletRed	Rhodamine	Mulberry
RedViolet	Fuchsia	Lavender	Thistle	Orchid
DarkOrchid	Purple	Plum	Violet	RoyalPurple
BlueViolet	Periwinkle	CadetBlue	CornflowerBlue	MidnightBlue
NavyBlue	RoyalBlue	Blue	Cerulean	Cyan
ProcessBlue	SkyBlue	Turquoise	TealBlue	Aquamarine
BlueGreen	Emerald	JungleGreen	SeaGreen	Green
ForestGreen	PineGreen	LimeGreen	YellowGreen	SpringGreen
OliveGreen	RawSienna	Sepia	Brown	Tan
Gray	Black	White		
GreenYellow	Yellow	Goldenrod	Dandelion	Apricot
Peach	Melon	YellowOrange	Orange	BurntOrange
Bittersweet	RedOrange	Mahogany	Maroon	BrickRed
Red	OrangeRed	RubineRed	WildStrawberry	Salmon
CarnationPink	Magenta	VioletRed	Rhodamine	Mulberry
RedViolet	Fuchsia	Lavender	Thistle	Orchid
DarkOrchid	Purple	Plum	Violet	RoyalPurple
BlueViolet	Periwinkle	CadetBlue	CornflowerBlue	MidnightBlue
NavyBlue	RoyalBlue	Blue	Cerulean	Cyan
ProcessBlue	SkyBlue	Turquoise	TealBlue	Aquamarine
BlueGreen	Emerald	JungleGreen	SeaGreen	Green
ForestGreen	PineGreen	LimeGreen	YellowGreen	SpringGreen
OliveGreen	RawSienna	Sepia	Brown	Tan
Gray	Black	White		

3 図

ワープロなどで文書の中に図を配置しようとするとき、適切な位置への調整は自力でしなければなりません。しかし、 $\text{T}_\text{E}_\text{X}$ では、図を最適(?)な場所に配置するための環境が用意されているので、これを利用します。

`figure` 環境は、図を張り込むための領域です。あくまで、「図を貼るための領域を確保する」ための環境ですから、実際の図を貼る命令ではありません。

3.1 図の挿入

まず、`figure` 環境は次のように利用します。

```
\begin{figure}
  実際の図を貼る命令
  \caption{図のキャプション}
  \label{ラベル}
\end{figure}
```

`figure` 環境の中には、実際の図を貼る命令のほかに図の説明をつける「キャプション」や参照するときに用いる「ラベル」を付記します。

実際の画像を張り込むためには、`\includegraphics` 命令を使用します。この命令は `graphicx` パッケージ内で定義されているので、プリアンプル部で

```
\usepackage{graphicx}
```

と読み込んでおきます。(よく似た名前でも `graphics` というものもありますが、性能的に `~x` の方が優れているようです。)

画像を貼るときは、貼りたい場所で

```
\includegraphics[サイズ]{画像ファイル名}
```

のように、引数に画像のファイル名を、オプション引数にどのくらいの大きさに貼るかサイズを指定します。サイズの指定方法は、`[width=3cm]` のように直接指定してもいいですが、

```
[width=0.4\textwidth]
```

のようにテキスト幅の `%`(上の例では `40%`) と指定した方が出力した原稿に対して視覚的にわかりやすいと

思います。

3.2 配置の指定

`figure` 環境で図を貼る領域を確保しますが、オプション引数でページのどの位置に領域を確保するか指定することができます。(あくまで、ページ内のどの部分かであって、大抵の場合実際に `figure` 環境を記述した場所から後ろに図が貼られます。) このオプション引数には、表 3.1 のようなものがあります。これらのオプショ

表 3.1: 図の配置指定

オプション引数	意味
<code>h</code>	その場所 (here) に配置する。
<code>t</code>	ページの上端 (top) に配置する。
<code>b</code>	ページの下端 (bottom) に配置する。
<code>p</code>	独立したページ (page) に配置する。

ンを以下のように指定します。

```
\begin{figure}[htbp]
  実際の図を貼る命令
  \caption{図のキャプション}
  \label{ラベル}
\end{figure}
```

$\text{T}_\text{E}_\text{X}$ は次の順序で図の配置場所を決定します。

- (1) 1 番目のオプション `h` に従い、`figure` 環境が記述されている位置に図を配置しようとする。
- (2) 適当な場所が確保できなければ、2 番目のオプション `t` に従い、ページの上端に図を配置しようとする。
- (3) それでも適当な場所が確保できなければ、3 番目のオプション `b` に従い、ページの下端に図を配置しようとする。
- (4) まだだめなら、最後のオプション `p` に従い、独立したページに図を配置します。

$\text{T}_\text{E}_\text{X}$ は自動的に領域を探して確保してくれますが、どうしてもその場所に貼りたいとか、ページの上部に貼りたい場合、つまりほかの場所に領域を確保することを許さない場合は

```
\begin{figure}[h!]
  実際の図を貼る命令
  \caption{図のキャプション}
  \label{ラベル}
\end{figure}
```

のように!マークを付記します。

3.3 図のキャプション

3.3.1 番号の前に付記される文字を変更する

図の番号はデフォルトでは「図 5」のように表示されますが、プリアンプル部に

```
\renewcommand{\figurename}{Fig.}
```

と記述することで「Fig.5」と変更することができます。

3.3.2 図 1: の部分を太字にする

図のキャプションの「図 1:」の部分を太字にするには、図 3.1 の命令をプリアンプル部に記述します。

ここで、上の記述をスタイルファイル内で記述する場合は`\makeatletter`、`\makeatother`を削除します。これは `caption` 命令に対して作用するので、表のキャプション番号も太字になります。

3.3.3 キャプション番号に節番号をつける

通常、キャプション番号は 1 から「図 1」のように連続的に付けられますが、

```
\renewcommand{\thefigure}{
  \thesection.\arabic{figure}}
\@addtoreset{figure}{section}
```

をプリアンプル部に記述することで「図 1.1」「図 1.2」、
「図 2.1」のように節毎に番号が付記されます。

3.4 図を複数枚並べて出力

`figure` 環境の中に複数の図を貼ることは、単純に `\includegraphics` 命令を続けて記述するだけです。しかし、`\caption` 命令を複数指定して図

を横に並べることは非常に困難です。(縦方向に並べる場合は、改行していけばいいので割と簡単です。)

通常、複数枚の図を横に並べて出力し、それぞれにキャプションを付記したいときは `minipage` 環境を使用します。T_EX で、横に複数枚の図を並べて出力したいときは以下の手順で記述します。

- (1) `figure` または `figure*` 環境を作る。
- (2) キャプションの数だけの `minipage` 環境を設置する。
- (3) 1 つのキャプションに対して 1 枚の図を貼る場合、各 `minipage` 内に 1 枚ずつ図を貼る
- (4) 1 つのキャプションに対して「(a)~」「(b)~」としたい場合、その枚数分 (2) で作った `minipage` 内に `minipage` を作り、そこで 1 枚ずつ図を貼る。
- (5) 適切な場所でキャプションをつける。

3.4.1 2 段抜き

`figure` 環境で確保される領域の横幅は本文 1 行の長さと同じです。(縦方向は貼って画像によって決まります。) このマニュアルのように本文を 2 段組みで組版している場合、図を横に 3 つ並べたいときなどでは、1 行の長さでは足りない場合がでてきます。このような場合には、`figure*` 環境を使用します。`figure*` 環境を使用すると、2 段分の幅の空間を確保できます。

本マニュアルでは、割と横に長いマクロ命令を記述するのに `figure*` 環境を使用しています。

3.4.2 minipage 環境

`minipage` 環境とは、その環境の中を、あたかも独立したひとつのページ(元のページに対して小さいページ)であるかのように扱うことができます。以下のように定義します。

```
\begin{minipage}{横サイズ}
  minipage の中
\end{minipage}
```

横サイズは `{5cm}` のように直接指定することができますが、`{0.45\textwidth}` のようにテキスト幅の 45% のように指定する方が各ページを均等に配置することができます。便利です。`minipage` を横に並べて設置する場合、`minipage` 間に若干 (`0.02\textwidth` 弱) のスペースが入ります。複数の `minipage` をテキスト幅一杯に均等に設

```

\makeatletter
\long\def\@makecaption#1#2{% \@makecaption を再定義します
  \vskip\abovecaptionskip
  \iftdir\sbox\@tempboxa{#1\hskip1zw#2}%
    \else\sbox\@tempboxa{\textbf{#1:} #2} % ここを変更する
  \fi
  \ifdim \wd\@tempboxa >\hsize%
    \iftdir #1\hskip1zw#2\relax\par
    \else \textbf{#1:} #2\relax\par\fi % ここを変更する
  \else
    \global \@minipagefalse
    \hbox to\hsize{\hfil\box\@tempboxa\hfil}
  \fi
  \vskip\belowcaptionskip}
\makeatother

```

図 3.1: 図のキャプションの番号を太字にする

置る場合は、この隙間も考慮に入れてください。縦方向は、minipage 内の内容によって自動的に決まります。

3.4.3 図を 2 枚横に並べて出力する

それぞれ別のキャプションをつける

2 段組みの片段に同じ大きさの図 (left.eps と right.eps) を 2 枚横に並べて出力する場合、図 3.2 のように記述します。その結果、図 3.3, 3.4 のようになります。

1 段組み, 2 段組みにかかわらずテキスト幅 `\textwidth` は同じになります。(2 段組みの場合、左段の左端から右段の右端です。) figure 環境内に設置する minipage の横幅は全体のテキスト幅の約 1/4 に設定します。

また、左右で図の大きさが違う場合は各々の minipage の横幅を調節します。

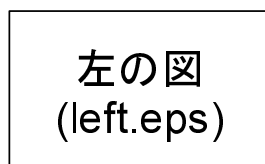


図 3.3: 左の図の説明

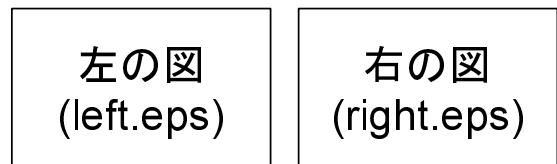


図 3.4: 右の図の説明

1 つのキャプションで (a), (b) とする

2 段組みの片段に同じ大きさの図 (left.eps と right.eps) を 2 枚横に並べて出力します。このとき、キャプションは 1 つとし、各画像の説明は (a), (b) とします。

このようにする場合、図 3.5 のように記述します。その結果、図 3.6 のようになります。



(a) 左の図の説明

(b) 右の図の説明

図 3.6: 図キャプション

3.4.4 図を横に 2 枚、縦に 2 枚並べて出力する

前節の図を横に 2 枚並べて出力する記述を 2 つ続けて書きます。TeX での改行は 1 行以上の空行なので、これらの記述の間に空行を挿入します。図 3.7 のように記述すると、図 3.8~3.10 のように出力されます。

```

\begin{figure}[htbp]
  \begin{center}
    \begin{minipage}{0.23\textwidth}
      \begin{center}
        \includegraphics[width=0.9\textwidth]{left.eps}
        \caption{左の図の説明}
      \end{center}
    \end{minipage}
    \begin{minipage}{0.23\textwidth}
      \begin{center}
        \includegraphics[width=0.9\textwidth]{right.eps}
        \caption{右の図の説明}
      \end{center}
    \end{minipage}
  \end{center}
\end{figure}

```

図 3.2: 図を 2 枚横に並べて出力する その 1

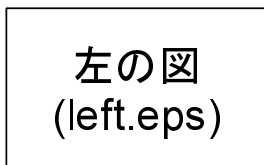
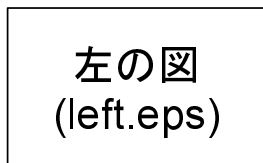


図 3.8: 左の図の説明



図 3.9: 右の図の説明



(a) 左の図の説明



(b) 右の図の説明

図 3.10: 図キャプション

3.4.5 図を 3 枚横に並べて出力する

2 段抜き (`figure*`環境) で同じ大きさの図を横に 3 枚並べ、それぞれキャプションをつける場合は図 3.11 のように記述すると、図 3.12~3.14 のように出力されます。

また、図 3.15(a), (b), 3.16 のように出力したい場合は、図 3.17 のように記述します。ここで、`\caption{右のキャプション}`の後の`{}`は 3 つの `minipage` の高さを揃えるための挿入した空白の行です。

3.5 フロートのレイアウト

`figure` 環境や `table` 環境は、図表を適切に配置する働きがあります。これらの環境は「ふわふわと場所を探してさまよう」ことから「フロート」と呼ばれています。

3.5.1 1 ページ内のフロートの数を変える

フロート環境が 1 ページ内にたくさんあると配置が困難になるため、`TEX` ではこの数を制限しています。デフォルトは表 3.2 のように設定されています。つまり、いくら小さいフロート環境であっても設定されている数以上のフロートを 1 ページ内に納めることはできません。

これらの値を変更するときは、`\setcounter` 命令を用いてプリアンブル部に

```

\setcounter{topnumber}{8}
\setcounter{bottomnumber}{8}
\setcounter{totalnumber}{8}
\setcounter{dbltopnumber}{8}

```

のように記述します。

```

\begin{figure}[htbp]
  \begin{center}
    \begin{minipage}{0.48\textwidth}
      \begin{center}
        \begin{minipage}{0.45\textwidth}
          \begin{center}
            \includegraphics[width=0.95\textwidth]{left.eps}
            (a) 左の図の説明
          \end{center}
        \end{minipage}
      \end{center}
    \end{minipage}
    \begin{minipage}{0.45\textwidth}
      \begin{center}
        \includegraphics[width=0.95\textwidth]{right.eps}
        (b) 右の図の説明
      \end{center}
    \end{minipage}
  \end{center}
\end{figure}

```

図 3.5: 図を 2 枚横に並べて出力する その 2

表 3.2: フロートの配置数

パラメータ	解説	デフォルト値
<code>\topnumber</code>	ページ上部に配置できるフロートの最大数	2
<code>\bottomnumber</code>	ページ下部に配置できるフロートの最大数	1
<code>\totalnumber</code>	ページ全体に配置できるフロートの最大数	3
<code>\dbltopnumber</code>	2 段組みのとき, ページ上部に配置できる 2 段抜きのフロートの最大数	2

3.5.2 本文とフロートの割合

`figure` 環境や `table` 環境の数とは関係なく版面のサイズやフロートのサイズなどによって, バランスが悪くなってしまうこともあります. そのため, $\text{T}_\text{E}\text{X}$ では本文とフロート環境の割合を設定するパラメータが用意されています.

デフォルトでは表 3.3 の値が設定されています.

これらの値を変更するときは, `\setcounter` 命令を用いてプリアンブル部に

```

\renewcommand{\topfraction}{1.0}
\renewcommand{\bottomfraction}{0.8}
\renewcommand{\dbltopfraction}{1.0}
\renewcommand{\textfraction}{0.0}
\renewcommand{\floatpagefraction}{1.0}
\renewcommand{\dblfloatpagefraction}{1.0}

```

のように記述します.

`\topfraction` は, 図や表がページ上部に占める部分


```

\begin{figure}[htbp]
  \begin{center}
    \begin{minipage}{0.23\textwidth}
      \begin{center}
        \includegraphics[width=0.9\textwidth]{left.eps}
        \caption{左の図の説明}
      \end{center}
    \end{minipage}
    \begin{minipage}{0.23\textwidth}
      \begin{center}
        \includegraphics[width=0.9\textwidth]{right.eps}
        \caption{右の図の説明}
      \end{center}
    \end{minipage}

    \begin{minipage}{0.48\textwidth}
      \begin{center}
        \begin{minipage}{0.45\textwidth}
          \begin{center}
            \includegraphics[width=0.9\textwidth]{left.eps} \\
            (a) 左の図の説明
          \end{center}
        \end{minipage}
        \begin{minipage}{0.45\textwidth}
          \begin{center}
            \includegraphics[width=0.9\textwidth]{right.eps} \\
            (b) 右の図の説明
          \end{center}
        \end{minipage}
      \end{center}
    \end{minipage}
    \caption{図キャプション}
  \end{center}
\end{figure}

```

図 3.7: 図を 2 枚横に並べて出力する その 3

の割合の上限を表します。上の例では 0.8 としてますので、ページの上から測って 80% までの部分が図だけで埋まっても良いという意味です。逆に、「2 枚の図を貼っ

たら 82% になる」場合には、2 枚目の図は次のページに追い出されてしまいます。

`\textfraction` というパラメータは逆に、文章が占

```

\begin{figure*}
  \begin{center}
    \begin{minipage}{0.32\textwidth}
      \begin{center}
        \includegraphics[width=0.95\textwidth]{left.eps}
        \caption{左の図のキャプション}
      \end{center}
    \end{minipage}
    \begin{minipage}{0.32\textwidth}
      \begin{center}
        \includegraphics[width=0.95\textwidth]{center.eps}
        \caption{真中の図のキャプション}
      \end{center}
    \end{minipage}
    \begin{minipage}{0.32\textwidth}
      \begin{center}
        \includegraphics[width=0.95\textwidth]{right.eps}
        \caption{右の図のキャプション}
      \end{center}
    \end{minipage}
  \end{center}
\end{figure*}

```

図 3.11: 図を 3 枚横に並べて出力する その 1

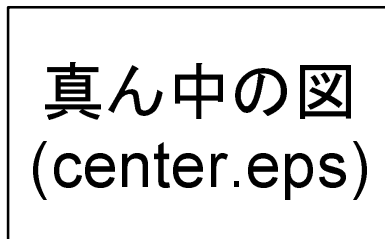
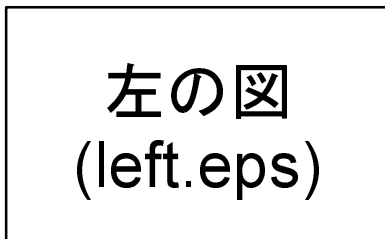


図 3.12: 左の図のキャプション

図 3.13: 真中の図のキャプション

図 3.14: 右の図のキャプション

有できるページの割合の下限を表します。上の例では、図がいっぱいあって「文章がページの 8%しか入らない」という場合には、「このページには文章を書かない」となります。ここを大きな数字にすると、図のみのページができてしまう事態がよく起きます。

3.6 図と表を並べて出力し、それぞれにキャプションをつける

通常、figure 環境や table 環境は 1 行の長さの幅を確保してしまうため、図と表は横に並べて出力することはできませんが、figure 環境で \caption 命令を使用すると「図 1:」と、table 環境では「表 1:」と出力されてしまうため、これらを同時に表示させることはできま



(a) 左の図の説明

(b) 右の図の説明

図 3.16: 右のキャプション

図 3.15: 左のキャプション

```

\begin{figure*}
\begin{center}
\begin{minipage}{0.63\textwidth}
\begin{center}
\begin{minipage}{0.48\textwidth}
\begin{center}
\includegraphics[width=0.95\textwidth]{left.eps}\\
(a) 左の図の説明
\end{center}
\end{minipage}
\end{center}
\end{minipage}
\begin{minipage}{0.48\textwidth}
\begin{center}
\includegraphics[width=0.95\textwidth]{center.eps}\\
(b) 右の図の説明
\end{center}
\end{minipage}
\caption{左のキャプション}
\end{center}
\end{minipage}
\begin{minipage}{0.3024\textwidth}
\begin{center}
\includegraphics[width=0.95\textwidth]{right.eps}\\
\caption{右のキャプション}
{}~
\end{center}
\end{minipage}
\end{center}
\end{figure*}

```

図 3.17: 図を 3 枚横に並べて出力する その 2

表 3.3: 本文とフロートの割合数

パラメータ	解説	デフォルト値
<code>\topfraction</code>	ページの上でフロートが占めることのできる割合	0.7
<code>\bottomfraction</code>	ページの下部でフロートが占めることのできる割合	0.3
<code>\textfraction</code>	本文とフロートの両方が混在するページのなかで本文がしめていなければならない割合	0.2
<code>\floatpagefraction</code>	フロートだけのページでフロートが最低限占めなければならない割合	0.5
<code>\dbltopfraction</code>	2 段組のとき、ページ上部で 2 段抜きフロートが占めることのできる割合	0.7
<code>\dblfloatpagefraction</code>	2 段組のとき、フロートだけからなるページでフロートが最低限占めなければならない割合	0.5

せん .

そこで、図 3.18 のマクロ命令をプリアンブル部に記述し、`figure` 環境内で表のキャプションを通常の `\caption` の代わりに `\tblcaption` とすることで正しく表示させることができます。(逆に `table` 環境内で、`\figcaption` 命令を使用して図のキャプションを書くこともできます。) ここで、上の記述をスタイルファイル内で記述する場合は `\makeatletter`、`\makeatother` を削除します。

図 3.19 のように記述すると、以下のように出力されます。

左の図 (left.eps)	表 3.4: 表のキャプション <table border="1" style="margin-left: 20px;"> <thead> <tr> <th></th> <th>サンプル A</th> </tr> </thead> <tbody> <tr> <td>$Q = 5$</td> <td>10</td> </tr> <tr> <td>$Q = 10$</td> <td>20</td> </tr> </tbody> </table>		サンプル A	$Q = 5$	10	$Q = 10$	20
	サンプル A						
$Q = 5$	10						
$Q = 10$	20						

図 3.20: 図のキャプション

```

\makeatletter
\newcommand{\figcaption}[1]{\def\@capttype{figure}\caption{#1}}
\newcommand{\tblcaption}[1]{\def\@capttype{table}\caption{#1}}
\makeatother

```

図 3.18: figure(table) 環境内で表 (図) のキャプションを正しく表示する

```

\begin{figure}[htbp]
  \begin{center}
    \begin{minipage}{0.23\textwidth}
      \begin{center}
        \includegraphics[width=0.9\textwidth]{left.eps}
        \caption{図のキャプション}
      \end{center}
    \end{minipage}
    \begin{minipage}{0.23\textwidth}
      \begin{center}
        \tblcaption{表のキャプション}
        \begin{tabular}{|c|c|}\hline
          & サンプル A \\\hline
          $Q=5$ & 10 \\\hline
          $Q=10$ & 20 \\\hline
        \end{tabular}
      \end{center}
    \end{minipage}
  \end{center}
\end{figure}

```

図 3.19: figure(table) 環境内で表 (図) のキャプションを正しく表示する 実際の例

4 表

図と同様に、 $\text{T}_{\text{E}}\text{X}$ は表を最適な場所に配置するための環境も用意しています。

`table` 環境は、表を貼り込むための領域を確保する環境です。単に「表を貼り込むための環境を確保する」だけですから、実際に表を貼り込んだりはしません。

4.1 表の挿入

`table` 環境は次のように使用します。

```
\begin{table}[htbp]
  \caption{表の説明}
  \label{ラベル}
  実際の表
\end{table}
```

また、`table` 環境は、オプション引数 (`[htbp]`) を指定することで、その表を出力する位置をある程度制御することができます。このオプション引数の種類と指定の仕方は `figure` 環境のものと同じですので、3.2 を参照してください。

なお、2 段組みの文書内で 2 段抜きの表を策定したい場合は `table` 環境の代わりに、`table*` 環境を用います。

4.1.1 要素の配置

実際の表は `tabular` 環境で作成します。表の基本的な書式は以下の通りです。

```
\begin{tabular}{列指定}
  表の中身
\end{tabular}
```

列指定では、表 4.1 に示す引数を列の数だけ並べます。

表 4.1: 表中のセルの配置

文字	要素の配置
l	左揃え
c	中央揃え
r	右揃え

実際に

```
\begin{center}
\begin{tabular}{lcr}
品名      & 数量 & 金額  & \\
チョコ    & 1    & 100 円 & \\
高級アイス & 30   & 15000 円 & \\
\end{tabular}
\end{center}
```

と記述すると、

品名	数量	金額
チョコ	1	100 円
高級アイス	30	15000 円

と出力されます。列指定

```
{lcr}
```

の l, c, r は左から 1 列目が左寄せ, 2 列目が中央, 3 列目が右寄せであることを表し底ます。表の中では, 列 (横方向) の区切りは &, 行 (縦方向) の区切りは \\ で行います。ただし, 罫線を引かない場合, 最下行に \\ は記述しません。

4.1.2 要素に長い文章を入れる

tabular 環境は要素の途中で折り返しをしないため, 表のセルに長い文章を入れたい場合, 表がページからはみ出してしまいます。

このような場合, TeX に列の幅を決定させるのではなく, こちらで列の幅を指定することにします。

列の幅を指定するには, tabular 環境の引数で l, c, r を指定する代わりに, p{30zw} などと記述します。p は段落 (Paragraph) という意味で, 引数にこの幅を指定します。この場合の 30zw は全角文字 30 文字分の幅を表します。p は折り返しを必要とする長い要素を前提としているため, 常に「左揃え」になります。

以下の例では, 右の列を全角文字 15 文字分 p{15zw} に設定しました。

桃太郎	むかしむかし あるところに おじいさんとおばあさんがお りました~
-----	---

4.2 罫線

4.2.1 縦罫線の引き方

表の要素間に縦の罫線を引くには, 縦の罫線を入れたい場所に対応する tabular 環境の列指定の中に | 記号を挿入します。また, || のように | 記号を 2 つ続けて指定すれば, 縦の二重線を引くことができます。

先ほどの表で

```
\begin{tabular}{|l||c|r|}
```

とすると, 各行は「縦の罫線」, 「左から 1 列目は左揃え」, 「縦の二重線」, 「左から 2 列目は中央揃え」~ のように出力されます。

品名	数量	金額
チョコ	1	100 円
高級アイス	30	15000 円

4.2.2 横罫線の引き方

横の罫線を引くには, 横の罫線を入れたい行間に \hline 命令を記述します。また, \hline 命令を 2 つ続けて記述することで, 横の罫線を二重に引くことができますが, 縦の罫線との交点が綺麗でないので 4.2.4 の書き方をします。

```
\begin{center}
\begin{tabular}{|l||c|r|} \hline
品名 & & 数量 & & 金額 & \\ \hline
チョコ & & 1 & & 100 円 & \\ \hline
高級アイス & & 30 & & 15000 円 & \\ \hline
\end{tabular}
\end{center}
```

と記述すると, 次のように出力されます。

品名	数量	金額
チョコ	1	100 円
高級アイス	30	15000 円

横の罫線を入れる場合は, 表の最下行のあとに \\ 命令を記述します。

4.2.3 部分的に横罫線を引く

\hline 命令を使用すると, 表の幅全体に横の罫線が出力します。場合によっては, 左から 3 列目だけ引きたかったり, 2 列目から 3 列目まで引きたいときもあります。このような場合, \hline 命令の代わりに, \cline 命令を使用します。

```
\cline{m-n}
```

とすることで、左から m 列目から n 列目まで、横の罫線を引くことができます。

4.2.4 横の二重線を引く

標準の $\text{T}_\text{E}_\text{X}$ における `tabular` 環境では、縦や横の罫線を引くことができます。 `hhline` パッケージを使用することで、`\hline` 命令を拡張した `\hhline` 命令を使用することができます。

`\hhline` 命令を使用するには、プリアンプル部で

```
\usepackage{hhline}
```

と記述し、`hhline` パッケージを読み込んだ上で、`\hline` 命令の代わりに、次の書式で使用します。

```
\hhline{引数}
```

引数には、`tabular` 環境の引数として指定した列指定 (`c` などの配置指定や `|` などの縦罫線指定) に対応するように、縦罫線や横罫線の種類を指定します。指定できる項目を表 4.2 に示します。

表 4.2: `hhline` 命令の引数

指定	解説
=	列の幅に相当する横の二重罫線
-	列の幅に相当する横の罫線
~	横の罫線を引かない
	横の罫線を貫く縦の罫線
:	横の罫線を貫かない縦の罫線
#	横の二重罫線と交わる縦の二重罫線
t	二重罫線の上罫線のみ (下罫線はなし)
b	二重罫線の下罫線のみ (上罫線はなし)

```

\begin{center}
\begin{tabular}{|l||c|r|} \hline
品名 & 数量 & 金額 \\ \hhline{|=#|=|}
チョコ & 1 & 100円 \\ \hline
高級アイス & 30 & 15000円 \\ \hline
\end{tabular}
\end{center}

```

と記述すると、次のように出力されます。

品名	数量	金額
チョコ	1	100円
高級アイス	30	15000円

`tabular` 環境の引数に指定されている列要素と、`\hhline` 命令の引数に指定されている罫線の種類は、次のように対応しています。

<code>\begin{tabular}{</code>	<code> </code>	<code>l</code>	<code> </code>	<code>c</code>	<code> </code>	<code>r</code>	<code> </code>
<code>\hhline{</code>	<code> </code>	<code>=</code>	<code>#</code>	<code>=</code>	<code> </code>	<code>=</code>	<code> </code>

4.3 セルを結合する

4.3.1 横方向のセルの結合

`tabular` 環境において、`\multicolumn` 命令を利用することで「横方向のセルを結合し見出しをつける」とか、「表の要素は左揃えで書きたいが、特定のセルだけ中央揃えにする」ことができます。

`\multicolumn` 命令は 3 つの引数を持ち、

```

\multicolumn{まとめるセルの数}{まとめたセルの要素の配置}{セルの中身}

```

で記述します。

これまで例で示してきた表の列の数は 3 です。この 1 行目に 3 つのセルを結合し、中央揃えで見出しを付ける場合、以下のように記述します。

```

\begin{center}
\begin{tabular}{|l||c|r|} \hline
\multicolumn{3}{|c|}{見積書} \\ \hline
品名 & 数量 & 金額 \\ \hhline{|=#|=|}
チョコ & 1 & 100円 \\ \hline
高級アイス & 30 & 15000円 \\ \hline
\end{tabular}
\end{center}

```

すると、次のように出力されます。

見積書		
品名	数量	金額
チョコ	1	100 円
高級アイス	30	15000 円

4.3.2 縦方向のセルの結合

縦方向のセルを結合する命令は用意されていません。そこで、 $\text{T}_{\text{E}}\text{X}$ では、

- (1) `\cline` 命令を使用して横罫線を消す
- (2) 必要に応じて `\raisebox` 命令を用いて、セル内の天地中央に文字を書く

ことをします。

`\raisebox` 命令、ベースライン (本文の下) から上にずらしたり、下にずらしたりして出力するための命令で、

```
\raisebox{上げ幅}[理論的な高さ][理論的な深さ]{テキスト}
```

と記述します。ここで、上げ幅には、テキストをどれだけ上に上げるかを指定します。そのため、負の値を指定すると下に下がります。

	サンプル A	サンプル B
サンプル C	1.5	2.5
	2.0	3.0

のように左の列の要素を縦に結合する場合、

```
\begin{center}
\begin{tabular}{|c|c|c|}\hline
& サンプル A & サンプル B \\ \hline
\raisebox{-1.75ex}[0pt][0pt]{サ ン プ ル C} & 1.5 & 2.5 \\ \cline{2-3}
& 2.0 & 3.0 \\ \hline
\end{tabular}
\end{center}
```

と記述します。この場合、「サンプル C」という文字を 1.5 の左側のセルに記述し、1.75ex 分下に下げています。

4.4 セルの背景に色をつける

論文投稿などでモノクロ印刷をする場合、薄い色の背景にするとこれがうまくハーフトーン化処理されて、あたかも網掛けをしたかようになります。そこで、表の中の特定のセルの背景に色をつけるために、`\cellcolor` 命令を定義します。

まず、プリアンプル部で `colortbl` パッケージを読み込みます。(別途 `colortbl.sty` を用意してください。)

```
\usepackage{colortbl}
```

続けて、`\cellcolor` 命令を定義します。

```
\newcommand{\cellcolor}[3]{\multicolumn{1}{|>\columncolor[named]{#1}}#2}{#3}}
```

(実際は改行しません。) `\cellcolor` 命令は 3 つの引数を持っており、表中の色を付けたいセルの部分で

```
\cellcolor{色の名前}{セルの環境引数}{セルの内容}
```

と記述します。

以下は、 3×3 の表の一番左上のセルの背景をシアン (Cyan) に変更した例です。

```
\begin{center}
\begin{tabular}{|c|c|c|}\hline
\cellcolor{Cyan}{c}{1} & 2 & 3 \\ \hline
a & b & c \\ \hline
A & B & C \\ \hline
\end{tabular}
\end{center}
```

のように記述すると、次のように出力されます。

1	2	3
a	b	c
A	B	C

色の名前は表 2.8 で定義されているものが使用できます。

5 数式

\TeX を作った Knuth 先生は数学者ということもあり、 \TeX は数式を非常に美しく出力することができます。

5.1 数式環境

\TeX では、数式はもちろん「ギリシャ文字」や「矢印」などの記号類の多くが数式環境でのみ出力されます。数式環境には大きく分けて次の 2 つがあります。

5.1.1 単純数式モード

単純数式モードは、本文中で変数や簡単な式を書くのに用います。数式の両側を $2x^2+3x-1$ のように数式を $\$$ 記号で囲むことで、本文中に $2x^2 + 3x - 1$ ように出力します。

5.1.2 別行立て数式モード

別行立て数式モードは、

$$2x^2 + 3x - 1$$

のように、独立した行に式を記述します。数式を 1 行だけ記述する場合は `equation` 環境、複数行を縦に並べて記述する場合は `eqnarray` 環境、複数行並べてすべての数式に式番号を付けない場合は `eqnarray*` 環境を用います。

上の場合、

```
\begin{eqnarray*}
2x^2+3x-1
\end{eqnarray*}
```

と記述しました。

5.2 空白

数式モードでは、 \TeX の命令の区切りを示す以外の半角空白は無視されます。つまり、 $x+1$ 、 $x_{\cup}+_{\cup}1$ 、 $_{\cup\cup}x+_{\cup\cup\cup}1$ もすべて同じ $x+1$ と出力されます。

数式中で空白を入れる場合は、表 5.1 の命令を使用します。

表 5.1: \TeX で使用できる空白

命令	空白	数式モード以外での利用
<code>\cup</code>	半角	可
<code>\quad</code>	全角 (クワタ)	可
<code>\qquad</code>	2 個のクワタ	可
<code>\,</code>	クワタの 1/6	可
<code>\></code>	クワタの 2/9	不可
<code>\;</code>	クワタの 5/18	不可
<code>\!</code>	クワタの -1/6	不可

5.3 添え字、べき乗

x_0 のような添え字は `_` 記号を用いて x_0 、 x^2 のようなべき乗は `^` 記号を用いて x^2 と記述します。また、これらは同時に使用することができます。例えば、 a_{i_j} と記述すると、 a_j^i となります。

添え字の文字が 2 文字以上の場合は、それらを `{ }` でくくります。 x^{10} と記述したいとき、この中カッコを入れ忘れ x^{10} と書くと x^{10} となってしまいます。

5.4 総和・総積

和の記号 \sum を出力するには `\sum` 命令を用います。

$$\sum_{k=1}^3 a_k = a_1 + a_2 + a_3$$

と記述すると、

$$\sum_{k=1}^3 a_k = a_1 + a_2 + a_3$$

と出力されます。

\sum のような特殊な記号については、別行立てモードと単純数式モードとでは大きさが異なるものがあります。`\sum` 命令を本文中で $\sum_{k=1}^3 a_k$ と書くと、 $\sum_{k=1}^3 a_k$ のように添え字の部分が小さく表示されてしまいます。これを別行立てのように $\sum_{k=1}^3 a_k$ と大きく表示する場合は 5.11 を参照してください。

5.5 分数

分数を書くには `\frac` 命令を用い、

`\frac{分子}{分母}`

のように記述します。例えば、`\frac{1-x}{1+x}`と書く

$$\frac{1-x}{1+x}$$

と出力されます。また、

`\frac{2}{1+\frac{x}{3}}`

のように分子(分母)中で`\frac`命令を使用することで、

$$\frac{2}{1+\frac{x}{3}}$$

のような繁分数式も書けます。

本文中で`\frac{1-x}{1+x}`と記述すると、 $\frac{1-x}{1+x}$ のように小さく出力されます。これを別行立てのように $\frac{1-x}{1+x}$ と出力したい場合は5.11を参照してください。

5.6 積分

積分記号 \int は`\int`命令で出力します。

`\int^b_a f(x) dx = F(b) - F(a)`

`\int^b_a f(x) \, dx = F(b) - F(a)`

と記述すると、

$$\int_a^b f(x) dx = F(b) - F(a)$$
$$\int_a^b f(x) \, dx = F(b) - F(a)$$

と出力されます。

上の2つの違いは、下の式の dx の前に、`\,`命令を挿入して少し空白を入れています。この方が美しく見えます。

2重積分などで積分記号を重ねるときは、必要な数だけ`\int`命令を記述します。

`\int\int_D f(x,y) dx dy`

`\int\!\!\!\int_D f(x,y) \, dx \, dy`

と記述すると、

$$\int \int_D f(x,y) dx dy$$
$$\int \!\!\!\int_D f(x,y) dx dy$$

と出力されます。

このとき、積分記号の間に`!`記号を(上の例では3つ)記述し、あいだを狭めた方が美しく見えます。

5.7 複数行の数式を揃える

複数の数式を並べて出力したい場合は、`eqnarray`環境を使用します。改行は`\\`命令を使用し、最後の行の式の後には記述しません。

```
\begin{eqnarray}
y=\sin x \\
y=\log_2 x \\
y=a_n x^n + \cdots + a_1 x + a_0
\end{eqnarray}
```

と記述すると

$$y = \sin x \tag{1}$$

$$y = \log_2 x \tag{2}$$

$$y = a_n x^n + \cdots + a_1 x + a_0 \tag{3}$$

と出力され、デフォルトでは右端で揃ってしまいます。

ここで、任意の演算子で揃えたい場合は、揃えたい演算子を`&`記号で囲みます。

```
\begin{eqnarray}
y&=\sin x \\
y&=\log_2 x \\
y&=a_n x^n + \cdots + a_1 x + a_0
\end{eqnarray}
```

と記述すると

$$y = \sin x \tag{4}$$

$$y = \log_2 x \tag{5}$$

$$y = a_n x^n + \cdots + a_1 x + a_0 \tag{6}$$

と出力されます。

5.8 長い数式を書く

2段組みで論文を執筆していると、1行の長さが短い
ため、長い数式を記述することができません。そこで、
&記号で数式を揃えることができることを利用して、数
式の途中で改行します。

```

\begin{eqnarray}
w &=& x+y+z \\
w^2 &=& x^2+2xy+2xz \nonumber \\
&& & +y^2+2yz+z^2
\end{eqnarray}

```

と記述すると

$$w = x + y + z \tag{7}$$

$$w^2 = x^2 + 2xy + 2xz + y^2 + 2yz + z^2 \tag{8}$$

と出力されます。ここで、3行目の+の前に{}が記述
されています。T_EX は数式中の空白を最適なものに調
節するため、この+は二項演算子の+であるとT_EXに
伝えるためです。

また、eqnarray環境では、すべての行の式に式番号
をつけるため、式を改行したときなど、式番号を付けたく
ないときは\nonumber 命令を使用します。

また、左辺が長く

$$\begin{aligned}
&u(x+h, y+k) + iv(x+h, y+k) \\
&= u(x, y) + iv(x, y) + (A+Bi)(h+ik) \\
&\quad + (\alpha(h, k) + i\beta(h, k))\sqrt{h^2+k^2}
\end{aligned}$$

のように出力したいときは、

```

\begin{eqnarray*}
\lefteqn{u(x+h, y+k) + iv(x+h, y+k)} \\
\hspace{1cm} \\
&=& u(x, y) + iv(x, y) + (A+Bi)(h+ik) \\
&& + (\alpha(h, k) + i\beta(h, k)) \\
&& \sqrt{h^2+k^2}
\end{eqnarray*}

```

のように、\lefteqn 命令を用います。この命令は、引
数に指定した原稿の長さ(幅)を0として扱います。1行
目で\lefteqn 命令を指定しただけだと、2行目以降の
数式が左端に揃えられてしまいます。そこで、2行目以
降が適度に字下げされるように、数式の代わりとなる適
当な長さ(ここでは1cm)の空白を1行目の左辺として
挿入しています。

5.9 行列

T_EX で、行列を書く方法はいくつかありますが、こ
こでは、最も基本的といえる array 環境について説明
します。

array 環境は、表を記述する tabular 環境とほぼ同
じです。以下のように記述すると、

```

\begin{eqnarray}
\left(
\begin{array}{ccc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23}
\end{array}
\right)
\end{eqnarray}

```

次のように出力されます。

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \tag{9}$$

array 環境では、\begin{array}に続く中カッコ{ }
の中に、各列について、中央揃えなら c、左揃えなら l、
右揃えなら r を書きます。各列は&で区切り、改行は\\
で行います。行列の大きなカッコは array 環境の外側

で、`\left`、`\right` 命令を用いて記述します。

5.10 場合分け

TeX で、条件によって値が変わる場合分けを書くには、「行列」を使う方法と、「cases 環境」を使う方法があります。例として、デルタ関数を記述します。

5.10.1 行列を使う場合

基本的には、それぞれの式、条件を行列の要素に割り当て、片側のカッコを削除します。以下のように記述すると、

```
\begin{eqnarray}
\delta (x)=
\left\{
\begin{array}{ll}
\infty & (x=0) \\
0 & (x\neq 0)
\end{array}
\right.
\end{eqnarray}
```

次のように出力されます。

$$\delta(x) = \begin{cases} \infty & (x = 0) \\ 0 & (x \neq 0) \end{cases} \quad (10)$$

5.10.2 cases 環境を使う場合

以下のように記述すると、

```
\begin{eqnarray}
\delta (x)=
\begin{cases}
\infty & (x=0) \\
0 & (x\neq 0)
\end{cases}
\end{eqnarray}
```

次のように出力されます。

$$\delta(x) = \begin{cases} \infty & (x = 0) \\ 0 & (x \neq 0) \end{cases} \quad (11)$$

行列の場合と同じように、要素は&で区切り、改行は\\で行います。cases 環境は amsmath パッケージで定義されているので、プリアンプル部で `\usepackage{amsmath}` と記述しなければなりません。

5.11 本文中で数式を大きく表示する

本文中で、分数や積分など小さく表示されてしまう数式を、別行立て数式モードで記述したのと同じ大きさで出力したいときは、`\displaystyle` 命令を用います。

本文中で分数を書く場合、本文中で

```
\frac{1-x}{1+x}
```

と記述した場合、

```
\frac{1-x}{1+x}
```

 と出力されますが、

\$の中で`\displaystyle`を

```
\displaystyle \frac{1-x}{1+x}
```

のように記述すると、

```
\displaystyle \frac{1-x}{1+x}
```

 のように出力されます。

5.12 カッコを大きく出力する

分数などで、以下のように

```
(\frac{x}{2})^2=\frac{x^2}{4}
```

普通にカッコで囲むと、

$$\left(\frac{x}{2}\right)^2 = \frac{x^2}{4}$$

とカッコが小さく出力されてしまいます。

ここで、`\left`、`\right` 命令を用いればカッコの大きさが自動で調節されます。

```
\left(\frac{x}{2}\right)^2=\frac{x^2}{4}
```

と記述すると,

$$\left(\frac{x}{2}\right)^2 = \frac{x^2}{4}$$

と出力されます.

ただし, `\left`, `\right` の数は同じでなければなりません. 片方だけカッコを付けたいときは,

```
\left(\frac{x}{2}\right.^2=\frac{x^2}{4}
```

$$\left(\frac{x}{2}^2 = \frac{x^2}{4}$$

のように, ピリオド(.) にします.

5.13 式番号の表示方法を変える

プリアンプル部に

```
\makeatletter
\renewcommand{\theequation}{%
\thesection.\arabic{equation}}
\@addtoreset{equation}{section}
\makeatother
```

と記述することで, 通常, 式番号が (1), (2), … となるところを, 1 節なら (1.1), (1.2), …, 2 節なら (2.1), (2.2), … に変更します. ここで, 上の記述をスタイルファイル内で記述する場合は `\makeatletter`, `\makeatother` を削除します.

通常は,

$$(x + y)^2 = x^2 + 2xy + y^2 \quad (12)$$

ですが, 上記のマクロ命令を追加することで

$$(x + y)^2 = x^2 + 2xy + y^2 \quad (5.13)$$

という表記になります.

6 EPS ファイル

BMP や JPEG 形式の画像はビットマップ形式で保存されています。これらは基本的に画素の集まりとして扱われているため、拡大/縮小といった解像度変更に対して画像は著しく劣化します。

それに対し、EPS ファイルは、PostScript によって記述されたベクトル形式の画像データを保存するためのファイル形式の一つです。印刷イメージなどの記述に使われる PostScript で画像ファイルを構成するための形式で、ベクタデータとビットマップデータを組み合わせた画像を保存することができます。ベクトル形式なので、線分であれば「端点」、円であれば「中心と半径」といった具合に情報を保存しているため、拡大縮小に対する画像劣化はありません。

6.1 EPS ファイルの作成方法

6.1.1 Illustrator を使う

EPS ファイルを作成するには、まず、通常通りに図を作成します。ここで、EPS はベクトル形式なので、画像のサイズをいくら縮小・拡大しようと、図のデータが損なわれることはありません。

ファイルを保存するときに、ファイルの種類を「Illustrator EPS (*.EPS)」に変更して保存します。

「保存」ボタンをクリックすると、図 6.1 のような EPS 形式オプション画面が表示されます。ここで、「フォントデータを含む(O)」にチェックを入れます。このチェックを入れないと、編集するときに設定したフォントの情報が EPS ファイル中に正しく反映されません。

6.1.2 WMF2EPS を使う

Plot32 で作成したグラフや PowerPoint で作成した図をクリップボードにコピーします。(通常は、変換したい図を選択して、コピー (Ctrl+C) します。)

WMF2EPS のメイン画面を開いて貼り付けます。すると図 6.2 のような「Select graph to paste」画面が出てきます。

ここで変換したい絵をとりあえず、WMF(Windows Meta File) か EMF(Extended Meta File) 形式で保存します。とりあえず EMF 形式で保存しておいてください。

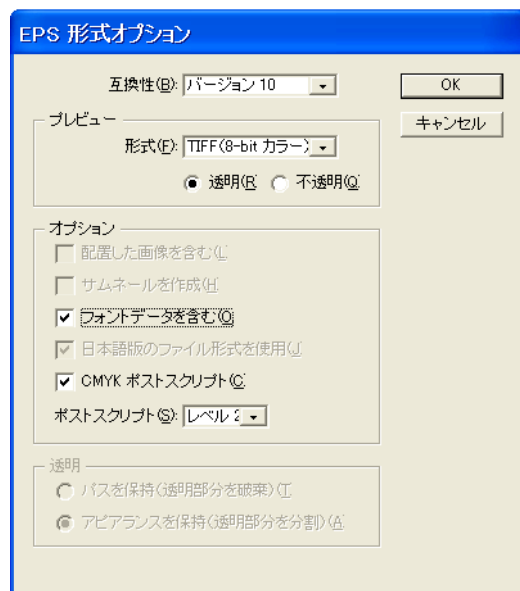


図 6.1: Illustrator 保存画面

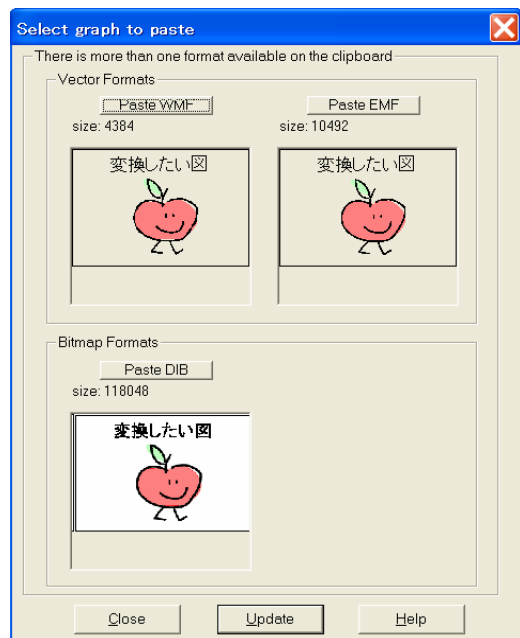


図 6.2: ペースト画面

次に、「EPS-Conversion」 - 「Current」をクリックすると、今表示されている画像が EPS 形式に変換されます。

7 PDF 変換

PDF ファイルとは、Adobe Systems 社によって開発された、電子文書のためのフォーマットです。作成した文書を電子的に配布することができ、相手のコンピュータの機種や環境によらず、オリジナルのイメージをかなり正確に再生することができます。

多くの場合、予稿集の原稿はその研究会独自のスタイルファイル(sty ファイル)を読み込んで作成し、PDF ファイルに変換して投稿します。

7.1 PDF ファイルを作る

PDF ファイルに変換するには、Adobe Acrobat と Acrobat Distiller(以下 Distiller) を使用します。

DVI ファイルから直接 PDF へと変換することはできません。まず始めに T_EX 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。

dvipsk を実行することで、原稿ファイルにエラーがなければ、DVI から PS ファイルに変換されます。この PS ファイルを図 7.1 に示したような、Distiller にドラッグ & ドロップします。PS ファイルに問題がなければ、PDF ファイルに変換されます。

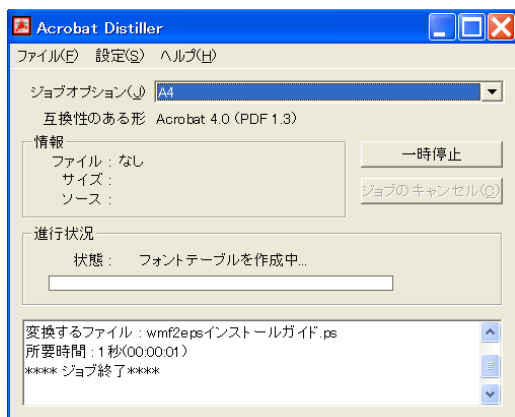


図 7.1: Acrobat Distiller 画面

出来上がりの用紙サイズや貼り付けた画像の画質は「設定(S)」 - 「ジョブオプション(J)」内で設定します。

7.2 レターサイズで変換するときの注意

日本では A4 サイズの用紙が一般的ですが、欧米ではレターサイズ(215.9mm × 279.4mm = 8.5 インチ × 11 インチ)が一般的です。そのため、この用紙サイズで提出しなければならない場合があります。

T_EX で用紙サイズ(\paperheight, \paperwidth)をレターサイズに設定した場合でも、PS ファイルが A4 サイズで作成されてしまい、Acrobat Distiller でページサイズを正しく指定しても、A4 サイズベースで作成されてしまうので、上部の余白が非常に狭くなってしまいます。

そこで、DVI ファイルから PS ファイルに変換するとき、用紙サイズをレターサイズで作成するように以下の引数を加えます。

```
-t letter
```

T_EX の処理に「M's TeX Helper2」を使用している場合、図 7.2 のような「M's TeX Helper2 の設定」を表示し、「TeX 処理設定」タグ内のプログラムパスから「DVI から PS へ変換するプログラム」を選択し、その引数の欄に上の引数を書き加えます。

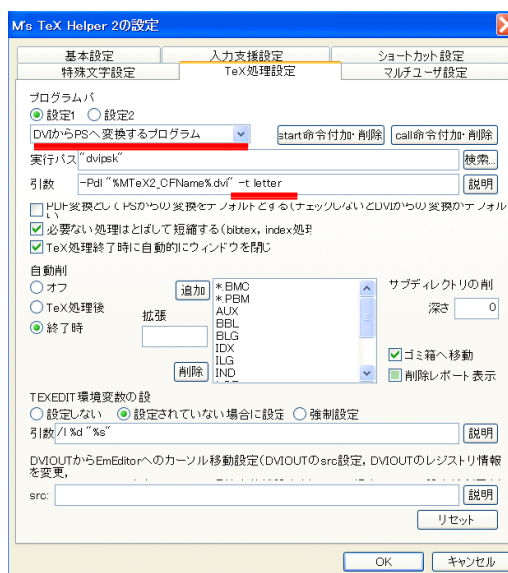


図 7.2: dvips の設定 (M's TeX Helper2 使用時)

7.3 PDF ファイルを作成するときの注意

PDF ファイルは相手のコンピュータの機種や環境に依存しないという理由からか、Windows の機種依存文

字である、「丸付き数字」や「全角のローマ数字」は表示できません。

TeX のコンパイル時は特にエラーメッセージも出ず、DVI ファイルでも普通に表示されるため、特に注意が必要です。TeX の本文中でローマ数字を使用する際は、和文・英文を問わず「半角英数字」を用いて “I”, “II”, …, “V” と書きます。