情報実習 I LaTeX 解説書

近畿大学理工学部情報学科

2021 年 5 月 26 日更新

目次

1 はじめに

LaTeX は理工系の論文・書籍などの出版に広く利用されている文書処理（組版処理）システムである．一般に使われている Word などのいわゆるワープロソフトとは使い勝手が異なるので，最初は戸惑うこともあるが，慣れてくると簡単に読みやすい文章を作成することができる．

情報実習 I では教科書 [1] を指定しているが，教科書は非常に細かに記述されているので，いきなり教科書を読んでも理解は難しいかもしれない．本書は教科書を読む前準備として概要を解説するものである．本書で概要を把握した上で教科書を読み進めていけば効率的に学修できるものと期待している．

なお，この文書自体も LaTeX で作成している．ソースファイルは GitHub[*1] で公開するので，興味があれば読んでみて欲しい．GitHub はソースコードなどを共有・管理するサービスである．オープンソースのソフトウェアの公開などに広く利用されている．

2 TeX と LaTeX

TeX は 1978 年に数学者の Donald E. Knuth によってリリースされた組版システムである[*2]．無料で使えるフリーソフトとしては配布されている．TeX を利用して組版するためには，文章とともに，文章の見栄え・体裁に関する指定を記述する．これは Web ページ（いわゆるホームページ）に使われている HTML 言語と同様であり，マークアップ言語と呼ばれている．

TeX 単体では使いにくい面があったので，より手軽に使用できるように機能拡張された文書処理システムが LaTeX である．現在では TeX 単体が利用されることは少なく，ほとんどの場合は LaTeX か LaTeX をもとに機能拡張された文書処理システムを使用することがほとんどである．

LaTeX の特徴は以下のようなものがあげられる．

- そのまま出版できるような綺麗な文書が作成できる．
- 数式を簡単に入力できる[*3]．
- 様々な OS で同じように利用できる．

後で解説するように，出力イメージを確認しながら編集するものではないので最初は戸惑うことも多いが，逆に，文章の内容に集中できる点が利点とも言えるので，慣れると使いやすくなる．

3 LaTeX による文書作成の流れ

文書作成の流れは以下のようになる．

(A) ソースファイルの編集
(B) タイプセットと PDF への変換

[*1] `https://github.com/habhit/LaTeXGuide`
[*2] 書籍などの紙面を構成する文字や図版を配置する工程のことを組版と呼ぶ．数学者である Knuth が組版システムを開発するに至った経緯など興味深い逸話がある．詳細は Wikipedia にも書かれている．
[*3] 最近では，MS Word などでも数式を簡単に入力できるようになってきているが，そこでは LaTeX にならった記法が使われるので，LaTeX に習熟していれば Word での数式入力にも役立つ．

(C)　プレビューや印刷

まず，ソースファイルを作成・編集する（A）．ソースファイルは文書本体と，見栄えを指定するコマンドが書かれたものである．LaTeX の場合のソースファイルは .tex という拡張子を付けるのが標準的である．ソースファイルの簡単な例を 4 節に示している．これは単純なテキストファイルであるので，好みのソフトウェアを使って編集していけば良い．

次に， タイプセット（組版）と PDF への変換を行う（B）．ここが LaTeX による処理の部分であり，ソースファイルを解釈して組版（文書や図版の書面への配置）を行う．組版の結果は PDF ファイルとして出力する[*4]．

最後に PDF ファイルを表示・印刷して内容を確認する（C）．最近の OS では，PDF ファイルを表示・印刷するためのソフトウェアは標準的にインストールされていることが多い．

ここに示した（B）や（C）の手順で問題が生じれば，（A）に戻ってソースファイルを修正することになる．

4　LaTeX ソースファイル

前節で述べた LaTeX ソースファイルの例を以下に示す．

```
\documentclass{jsarticle}

\begin{document}

 こんにちは \LaTeX !!

\end{document}
```

ソースファイルについて，まずは以下の点だけは覚えておいて欲しい．

- \documentclass　はこれから書く文書の種類・スタイルなどを指定するコマンドである．当面はあまり意識しなくても良い．
- \documentclass　から \begin{document} の間のことを**プリアンブル**（前書き）と呼ぶ．プリアンブルには追加機能を読み込むコマンドなど，文書全体に影響を及ぼすコマンドを書く．
- \begin{document} から \end{document} の間に文章の本文と，その構成や見栄えを指定するコマンドを記述する．

ソースファイルをタイプセットするための手順や，うまく行かなかった場合の処置などについては実習の解説スライドを参照のこと．上で示したソースファイルをタイプセットすると，以下のような文だけの文書が得られる．

こんにちは LaTeX!!

[*4] PDF ファイルは文書を配布する形式として広く利用されている．LaTeX による組版結果は dvi ファイルとして出力されているが，dvi ファイルの内容を確認するには特別なソフトが必要であるので，最近では PDF に変換することが多い．

5　コマンドリファレンス

今後 LaTeX で文書作成する際に便利なように，実習で取りあげる LaTeX コマンドについて以下にまとめておく．それぞれのコマンドの詳しい使い方はここでは解説しないので，不明点があれば教科書 [1] を確認すること．

なお，特に注記しない限り，以下のコマンドはすべて文章の本文を記述するときに用いるので，4 節で示したソースファイルの例では `\begin{document}` から `\end{document}` の間に記述する．

5.1　特殊文字

LaTeX では特別な意味をもつ文字をそのまま出力したいときには表 1 のように書く．

表 1　特殊文字の例

入力	出力
`\#`	#
`\$`	$
`\%`	%
`\&`	&
`\_`	_
`\{`	{
`\}`	}

5.2　改行

LaTeX　のソースファイルでは改行が一つあっても結果に影響を与えない．表 2 にあるように，改行を二つ連続して書く（何もない行をつくる）と改行されて，後に続く文章の冒頭は字下げが行われる（インデントありで改行される）．つまり，空行は段落の区切りの意味ももっている．むやみに使うと読みにくい文書になるので注意すること．

通常，本文の中で強制改行は行わず，段落の区切りでの改行（インデントあり）のみが行われる．`\\` で強制改行が可能だが，タイトル・表・数式など特別な場合のみに利用する．

表 2　改行

入力	出力
空行（何もない行）	改段落（インデントあり改行）
`\\`	インデントなし改行
`\noindent`	インデントの解除

5.3 書体・文字サイズ

文章中で強調などの必要があれば表 3 のように書体や文字サイズを変更することができる．ただし，これらを多用すると効果がなくなって逆に読みづらくなるので注意すること．特に，本文中で文字サイズを変更すると著しく読みづらくなるので，基本的には本文中では文字サイズを変更しないようにする．なお，特に日本語を使っているときには，斜体などの書体が無い場合もあるので注意する．

表 3　書体・文字サイズの変更

入力	出力	説明
`\textrm{sample}`	sample	ローマン（基本）
`\textbf{sample}`	**sample**	ボールド（太字）
`\textit{sample}`	*sample*	イタリック（斜体）
`{\tiny sample}`	sample	
`{\footnotesize sample}`	sample	
`{\large sample}`	sample	
`{\Large sample}`	sample	
`{\LARGE sample}`	sample	

5.4 環境

`\begin{何々}` と `\end{何々}` によって要素（文章や数式など）を囲むコマンド，あるいは囲まれている領域を環境と呼ぶ．「何々」の部分に指定するものによって働きが変わってくる[*5]．

5.4.1 配置に関する環境

例えば，中央揃えをした場合は，

```
\begin{center}
中央揃えしたいもの
\end{center}
```

とソースファイルに書けば，

中央揃えしたいもの

と出力される．他には表 4 に示すような環境がある．

5.4.2 箇条書きに関する環境

箇条書きを適切に使うと分かりやすい文章になる．箇条書きに関する環境を表 5 に示す．

itemize 環境を使う場合は，

[*5] 4 節で示した例にある本文を記述する部分も document 環境である．

表 4　配置に関する環境

環境	内容
quote 環境	左右の余白（引用に使う）
center 環境	中央揃え
flushleft 環境	左寄せ
flushright 環境	右寄せ

表 5　箇条書きに関する環境

環境	内容
itemize 環境	箇条書き（各項目の冒頭に同じ記号がつく）
enumerate 環境	番号付き箇条書き（各項目の冒頭に通し番号がつく）
description 環境	見出し付き箇条書き（各項目の冒頭に任意の見出しがつく）

```
\begin{itemize}
\item テキスト 1
\item テキスト 2
\item テキスト 3
\end{itemize}
```

とソースファイルに書けば,

- テキスト 1
- テキスト 2
- テキスト 3

と出力される.

enumerate 環境を使う場合は,

```
\begin{enumerate}
\item テキスト 1
\item テキスト 2
\item テキスト 3
\end{enumerate}
```

とソースファイルに書けば,

1. テキスト 1
2. テキスト 2
3. テキスト 3

と出力される.

description 環境を使う場合は,

```
\begin{description}
\item[見出し 1] テキスト 1
\item[見出し 2] テキスト 2
\item[見出し 3] テキスト 3
\end{description}
```

とソースファイルに書けば,

見出し 1　テキスト 1
見出し 2　テキスト 2
見出し 3　テキスト 3

と出力される．\item コマンドの使い方が先の二つの環境とは異なることに注意する．

5.5　コメント行

コメント行はタイプセット時には無かったものとして処理される．コメント行をうまく利用すると効率的に文書作成を行える．例えば，以下のような用途がある．

- あとでソースファイルをみたときに参考になるような記述を残しておきたい場合
- タイプセットで何か不具合が発生した際にソースファイルにある不具合の原因を絞り込む場合[*6]

ある部分をコメントにすることを「コメントアウト」と呼ぶ．コメントアウトの方法には大きく分けて二つある．

5.5.1　一つの行のコメント

行頭に % コマンドを付けるとその行はコメントとみなされる．例えば,

```
あいうえお
% かきくけこ
さしすせそ
```

とソースファイルに書けば,

あいうえおさしすせそ

と出力される．なお，ここで,「あいうえおかきくけこ」と一行で表示されているのは，5.2 節で述べた内容と一致しているので，確認されたい．

[*6] タイプセットでエラーが発生してその原因を探りたいとき，ある部分をコメント行にしたときにタイプセットのエラーが発生しなくなれば，コメント行にした部分が原因である可能性が高い．このようにしてエラーに対処する方法はプログラミングでも有効である．

5.5.2 複数行のコメント

\iffalse と \fi で挟んだ行はコメントとみなされる．例えば，

```
あいうえお
\iffalse
かきくけこ
さしすせそ
\fi
たちつてと
```

とソースファイルに書けば，

```
あいうえおたちつてと
```

と出力される．

5.6 タイトル

表 6 に示すコマンドを使えば，レポートなどのタイトルを簡単に作成することができる．まず，\title{}，\author{}，\date{}コマンドを使ってタイトル・著者名・日付を指定したあと，\maketitle コマンドを実行してタイトルを出力する．

表 6　タイトルに関するコマンド

コマンド	内容
\title{}	タイトルの指定
\author{}	著者名の指定
\date{}	日付の指定
\maketitle	タイトル・著者名・日付の出力

例えば，

```
\title{情報実習 I レポート}
\author{情報太郎}
\date{2019 年 5 月 1 日}
\maketitle
```

とソースファイルに書けば，

表 7　階層的な見出しを付けるためのコマンド

コマンド	内容
\part{}	部
\chapter{}	章
\section{}	節
\subsection{}	小節（項）
\subsubsection{}	小小節（目）
\paragraph{}	段落
\subparagraph{}	小段落

情報実習 I レポート

情報太郎

2019 年 5 月 1 日

と出力される．なお，タイトル作成については次の点も留意しておくと良い．

- タイトル・著者名が長くなってしまった場合は \\ で強制改行して見やすくすると良い（5.2 節参照）．例えば，\title{**タイトルの一行目 \\ タイトルの二行目**}のように書く．
- 著者が複数いる場合は，\and コマンドで区切って記述する．例えば \author{**情報太郎 \and 近大花子**}のように書く．
- 日付を指定するときには \today という特別なコマンドがあり，\date{\today} とすればタイプセットを行った日が自動的に指定される．

5.7　節番号

レポートなどの技術文書ではその構造が重要になる．文書の構造については教科書 [1] の第 3 章の 3.1 節で説明されているのでそちらを参照すること．文書の構造を分かりやすく示すために，表 7 に示すコマンドを使って階層的な見出しをつける．

表 7 に示すコマンドはいつでも利用できるわけではなく，作成している文書のスタイルによって利用可能なコマンドが決まる．例えば，「部」「章」は書籍のような大規模な文書を執筆する際に利用し，レポートなどの場合には使用しない．当面は \section{} や \subsection{} を中心に使用し，もし必要があればさらに細かな階層のコマンドを用いるようにする．

もちろん，この文書自体も \section{} や \subsection{} を使用している．例えば，5 節の見出しを付けるためには

```
\section{コマンドリファレンス}
```

と，5.7 節の見出しを付けるためには，

```
\subsection{節番号}
```

とソースファイルに記述している．このように，節や小節の見出しのみを記述すれば，LaTeX が連番を割り当ててくれる．

5.8 表の作成

実験や調査の結果を分かりやすく示すため表を用いると良い場合がある．表の作成のため多くのコマンドが用意されており，それらを駆使すると複雑な表を作成できるが，まずは単純なものをマスターするようにすれば良い．

まず，簡単な例を以下に示す．

```
\begin{table}[!h]
\caption{ls コマンドの使い方と実行結果}
\label{tab:ls}
\begin{center}
\begin{tabular}{|c||c|}\hline
コマンド & 実行結果 \\ \hline
\verb*|ls| & カレントディレクトリのファイル一覧を表示 \\ \hline
\verb*|ls ^^ef^^bd^^b0 l| & ‘‘説明’’ \\ \hline
\verb*|ls - al| & ‘‘説明’’ \\ \hline
\end{tabular}
\end{center}
\end{table}
```

のようにソースファイルに記述すると，表 8 のような表が出力される．これを順を追って説明していく．

表 8　ls コマンドの使い方と実行結果

コマンド	実行結果
ls	カレントディレクトリのファイル一覧を表示
ls␣^^ef^^bd^^b0␣l	“説明”
ls␣-␣al	“説明”

- 最初の行の \begin{table} と最後の行の \end{table} の対で table 環境を使ってる．table 環境は表の配置を決め，その中に見出しや表の本体を記述する．
- \begin{table} のあとについている [!h] は表の配置を指定するオプションである．配置を指定するオプションには表 9 のような種類がある．これらは複数指定することができ，最初に指定した方が優先される．例えば htbp と指定すれば，まず h による配置が試みられ，うまく配置できなければ次に t という順に処理が行われる．表や図を配置する場合は，文章中の図表の割合などの基準と照らし合わせて LaTeX が適切な場所を選択するので，必ずしもユーザーの指定通りにはいかない．特に最初はもど

表 9　表や図の配置を指定するためのオプション

オプション	内容
h	この場所（here）
t	ページ上端（top）
b	ページ下端（bottom）
p	表や図だけのページを用意してそこに出力
!h	できるだけここに出力

表 10　表中の列の出力形式を指定するためのオプション

オプション	内容
l	項目が左寄せされる列
c	項目が中央寄せされる列
r	項目が右寄せされる列
|	列と列の間に一本縦線を引く
||	列と列の間に二本縦線を引く

かしい思いをすることもあるが，5.9 節で述べる相互参照が正しく行われていれば，まずは問題ないと考えて良い．

- \caption{} は表の見出しを指定する．表では見出しは表本体の上部に書き，図では見出しは図本体の下部に書く．
- \label{} は相互参照を用いるときに使用するラベルを指定する．詳しくは 5.9 節で述べる．
- center 環境は 5.4.1 節で説明したとおり，中央に揃える環境である．
- \begin{tabular} と \end{tabular} の対に囲まれている tabular 環境を用いて表本体を記述する．
- \begin{tabular} コマンドについているオプション {|c||c|} は表に含まれる列の数と形式を指定している．c は項目が中央寄せされる列を，| は列の間の縦線を示している．表 8 の出力内容と照らし合わせて確認すること．他にも多くのオプションがあるが，代表的なものを表 10 に示している．
- tabular 環境の中に記述されているコマンドの意味は次のとおりである．

  \hline：横線を出力する

  \\：改行

  &：列の区切り

  一行に含まれる列の数は冒頭の \begin{tabular} のオプションで指定した列の数に一致しなければならないので注意する．
- 表の中に記述されている \verb|| と \verb*|| は文字をそのまま表示したいときに用いるコマンドである．|| に挟まれた文字列がそのまま出力される．\verb|| と \verb*|| はほぼ同じであるが，後者は半角スペースを ␣ のように明示する

5.9 相互参照

5.8 節で説明した表や，次の 5.10 節で説明する図を挿入したときは，本文の中でもそれについて言及・説明するのが基本である．基本的には，どこからも言及されない図・表は存在しないと考えて欲しい．表や図について言及するときには，これまでの説明でも出てきているように，「表 10 のように・・・」のような形で，表や図の番号を示して参照する．これは表や図に限った訳ではなく「5.8 節で・・・」のように節などにも当てはまる．

表・図・節などを参照するときには，その番号を手入力する必要はなく，自動的に適切な番号を出力する機能が備わっている．そのためには，二つのコマンド `\label{}` と `\ref{}` を対で利用する．まず，`\label{}` を使ってあらかじめ参照先（表・図・節）にラベルを付けておく．例えば 5.8 節で示したソースファイルでは，

```
\label{tab:ls}
```

として `tab:ls` というラベルをつけている[*7]．

ラベルを付けておいたものを参照するときは，

```
表 \ref{tab:ls} に示すように・・・
```

のように `\ref{}` コマンドを用いて記述しておけば，タイプセットを実行するときに自動的に数値に置き換わって，「表 8 に示すように・・・」と出力される．

5.10 図の挿入

説明のために必要な模式図や写真は「図」として挿入する．図を挿入するためには様々な方法があるが，ここでは画像ファイル（PNG ファイル）を挿入する方法を説明する．

標準では画像ファイルを挿入できないため，追加機能として `graphicx` パッケージを用いて画像ファイルを挿入する．`graphicx` パッケージを利用するためにはプリアンブル（4 節参照）に

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

と記述する．その上で，本文中に次のように記述すれば画像による図が挿入される．

```
\begin{figure}[!h]
\begin{center}
\includegraphics[scale=0.6,clip]{ls.png}
\caption{ls,ls - al コマンド実行の様子}
\label{fig:ls}
\end{center}
\end{figure}
```

[*7] ラベルは半角の英字・数字であれば何でもよいが，表は tab:，図は fig: を頭につける，などのルールを設けておけば混乱しなくて良い．

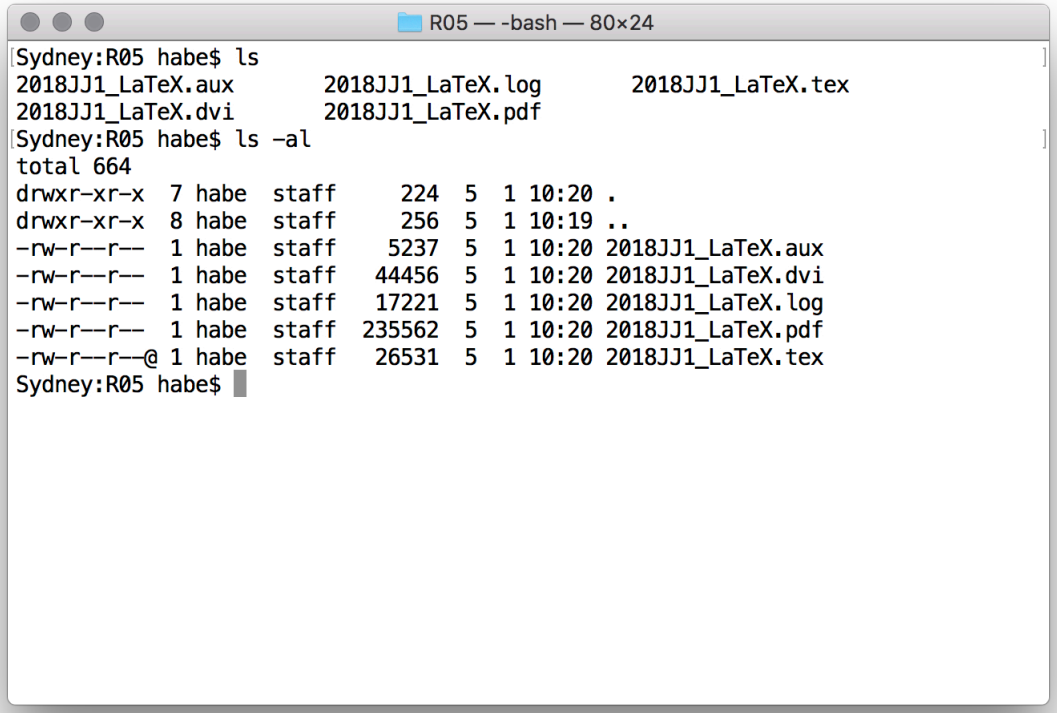

図 1　ls,ls - al コマンド実行の様子

これを順を追って説明していく．

- 最初の行の \begin{figure} と最後の行の \end{figure} の対で figure 環境を使ってる．table 環境と同じように，図の配置を決め，その中に画像など図の本体や見出しを記述する．
- \begin{figure} のあとについている [!h] は表の配置を指定するオプションである．配置を指定するオプションは表と同じであり，表 9 のような種類がある．
- center 環境は 5.4.1 節で説明したとおり，中央に揃える環境である．
- \includegraphics が画像を挿入するコマンドである．ここで指定されているオプションの意味は次のとおりである．

  scale=0.6：元の画像サイズの 0.6 倍で出力する．タイプセットしたあとの大きさを確認して適切な数値にすること．小さくて図中の文字が見えないようなことは厳禁である．

  clip：はみ出した分を切り取る．通常は指定しておいた方が良い．

  {ls.png}：{}で画像ファイルを指定する．当然ながらファイル名を間違って存在しないものを指すとエラーになるので注意する．なお，ここに相対パスでファイル名を書くと，コマンドを実行したディレクトリからみたものとして解釈される．
- \caption{} は図の見出しを指定する．表では見出しは表本体の上部に書き，図では見出しは図本体の

下部に書く．

- `\label{}` は相互参照を用いるときに使用するラベルを指定する．詳しくは 5.9 節で述べている．

画像ファイルを準備する方法は様々であるが，実習資料ではスクリーンショットをとる方法を解説している．

5.11 数式の記述

2 節で述べたように，数式を簡単に入力できるのが LaTeX の大きな特長の一つである．最近は，MS Word などでも数式を簡単に入力できるようになっているが，LaTeX の記法にならっていることが多いので，LaTeX に習熟していれば様々な場面でも役立つ，

標準でもある程度の数式を記述できるが，$\mathcal{AMS}$- LaTeX パッケージ[*8]には様々な便利な機能が備わっているので，以降では $\mathcal{AMS}$- LaTeX を含めて基本的な利用方法を解説していく．まず，$\mathcal{AMS}$- LaTeX パッケージを利用するための「おまじない」をプリアンブル（4 節参照）に

```
\usepackage{amsmath}
```

と記述しておく．

5.11.1 インライン数式とディスプレイ数式

LaTeX で数式を記述する場合には，「インライン数式」と「ディスプレイ数式」の二つの方法がある．

インライン数式とは，文中で数式を扱うための方法であり，文中で式を $ で囲むように記述する．例えば，ソースファイルで `$ax+b$` と書いた場合，出力結果は $ax + b$ となる[*9]．これに対して，数式モードを使わずに `ax+b` と書いた場合には ax+b となり出力結果が異なっている．数式モードで出力すると斜体で出力されるが，数式モードを使わない場合はローマン体で出力されるので，くれぐれも間違えないようにすること．

ディスプレイ数式とは，`equation` や `align` 環境を用いて，別行立てで数式を記述する方法である．例として，`align` 環境を用いた数式を以下に記述する．

```
\begin{align}
f(x) = a x^{2} + b
\end{align}
```

このように記述すると，以下のように出力される．

$$f(x) = ax^2 + b \tag{1}$$

この例では，`x^{2}` としている部分は出力例のように `x` の「肩にのせる」ものを指定している．べき乗などを記述する際に用いる．

5.11.2 verbatim 環境

数式とは直接関係ないが，`verbatim` 環境はソースファイルに記述した内容をそのまま出力する．例えば，

[*8] アメリカ数学会（American Mathematical Society）向けに開発されたものであるので，数学者向けの機能が備わっている．

[*9] 細かいが $ で囲んだ部分の前後には半角スペースを入れた方が見栄えが良い．

```
\begin{verbatim}
\begin{align}
f(x)=ax^{2}+b
\end{align}
\end{verbatim}
```

と記述すれば,

```
\begin{align}
f(x)=ax^{2}+b
\end{align}
```

と表示される．ここでは LaTeX　のソースファイルに記述する内容を示すために用いているが，今後作成するプログラムのソースファイルをレポート中に記載する場合などにも利用できる．また，verbatim の代わりに verbatim* 環境を用いれば半角スペースを記号に置き換えたものが表示される．ぜひ試してみるとよい．

5.11.3　数式で用いる文字・記号・演算子

数式を記述するために必要な文字・記号・演算子を表 11 にまとめる．記号などは英語での呼び方がコマンドになっている場合が多く，馴染みがないものもあると思うが，必要に応じてマニュアルを参照しながら書いていけば良い．表 11 に挙げているのは一部分である．たとえば [2] のような Web サイトにも様々なコマンドがまとめられているので，随時参考にすればよい．

表 11　数式で用いる文字・記号・演算子

ギリシャ文字		関係・二項演算子		三角関数・数学記号	
コマンド	出力	コマンド	出力	コマンド	出力
\alpha	α	\times	$\times$	\sin	sin
\beta	β	\div	$\div$	\cos	cos
\gamma	γ	\pm	$\pm$	\tan	tan
\delta	δ	\ne	$\neq$	\log	log
\pi	π	\leq	$\leq$	\lim	lim
\mu	μ	\geq	$\geq$	\exp	exp
\sigma	σ	\in	$\in$	\min	min
\omega	ω	\notin	$\notin$	\max	max
		\cap	$\cap$	\arg	arg
		\cup	$\cup$		

5.11.4　行列

線形代数などでよく扱う行列も LaTeX を使えば簡単に記述でき，美しく組版できる．LaTeX 標準の array 環境を使っても良いが，ここではより高機能な pmatrix 環境を紹介する．

```
\begin{align}
\begin{pmatrix}
\alpha & \beta
\\
\gamma & \delta
\end{pmatrix}
\end{align}
```

このように記述すると，以下のように出力される．

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \tag{2}$$

行列の記述方法は 5.8 節で説明した表の記述方法と似ており，\\ が改行を示し，& が列の区切りを示している．ソースと出力例を照らし合わせて見て欲しい．

もう少し複雑な例としては，

```
\begin{align}
\begin{pmatrix}
\alpha & \beta
\\
\gamma & \delta
\end{pmatrix}
\begin{pmatrix}
x_{1}
\\
x_{2}
\end{pmatrix}
=
\begin{pmatrix}
\alpha x_{1} + \beta x_{2}
\\
\gamma x_{1} + \delta x_{2}
\end{pmatrix}
\end{align}
```

とすれば，

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \alpha x_1 + \beta x_2 \\ \gamma x_1 + \delta x_2 \end{pmatrix} \tag{3}$$

のように出力される．pmatrix 環境で表現される行列が一つの記号のようになっていることに注意されたい．

5.11.5 分数・平方根

これまでに紹介したものの他に，数式では分数や平方根もよく登場する．

```
\begin{align}
g(x) = \frac{a}{b} x^{2} + \sqrt{c}
\end{align}
```

このように記述すると，以下のように出力される．

$$g(x) = \frac{a}{b}x^2 + \sqrt{c} \tag{4}$$

`\frac{a}{b}` は分数を記述するコマンドである．出力例を見ればわかるように `\frac` の次に分子，その次に分母を記述する．`\sqrt{c}` は平方根を記述するものである．`\frac{a}{b}` と `\sqrt{c}` のどちらも括弧の名に記述した数式が大きくなってもバランスよく数式が配置されるように自動調整される．

5.12 参考文献リスト

レポートや論文などを書くときには，まったく何も参考にせずに，ゼロから書き上げることはまず起こりえない．必ず何らかのを文献を参考にし，そこから得た知識を利用して，自らの考察・提案をまとめていくことになる．どの部分を参考にし，どの部分が独創的な部分なのかを明確にするためにも，参考文献リストを明示し，それが本文のどの部分に関与しているのかを明らかにしておく必要がある．

参考文献リストを作成するために，LaTeX には様々な便利な機能が備わっているが，ここではもっとも単純なものを紹介しておく[*10] ．

参考文献リストは文章の末尾に以下のように記述しておく．

```
\begin{thebibliography}{9}
   \bibitem{TeXText} 渡辺徹，好き好き \LaTeXe 初級編，2006．
   \bibitem{Kagishippo} 「物理のかぎしっぽ」\TeX に関するメモ，2021 年 5 月 17 日閲覧．
\end{thebibliography}
```

この例は本書の末尾にある参考文献リストとほぼ同じもの出力するためのソースである[*11]．簡単にその内容を説明する．

- `\begin{thebibliography}` のあとにある `{9}` の数値は参考文献数の桁数を示している．もし参考文献の数が一桁であれば 9 ，二桁であれば 99 のように記述する．
- `\bibitem` で始まる項目がそれぞれの参考文献を指している．`\bibitem` は直後に参照キーを指定する．本文ではこの参照キーを使って参考文献を指し示す．

このような参考文献リストを用意したあとで，本文中では `\cite` コマンドを使って参考文献について言及する．たとえば，

[*10] BibTeX というツールを使えば，あらかじめ用意しておた文献データベースから自動的に整形して参考文献リストを作成できる．
[*11] Web アドレスを書くと長くなるので省略している．

```
情報実習 I では教科書\cite{TeXText}を指定しているが，教科書は非常に細かに記述されているので，
いきなり教科書を読んでも理解は難しいかもしれない．
```

とソースファイルに書けば，

情報実習 I では教科書 [1] を指定しているが，教科書は非常に細かに記述されているので，いきなり教科書を読んでも理解は難しいかもしれない．

と出力される．参照キーを用いれば，それが何番の参考文献に相当するかは LaTeX が自動的に判断して番号に置換される．

なお，参考文献リストで Web ページのアドバイスを示すときには `url` パッケージを使えば便利である．プリアンブルに

```
\usepackage{url}
```

と書いておけば，

```
\url{https://www.kindai.ac.jp/science-engineering/}
```

という命令をソースファイルに書けば，

```
https://www.kindai.ac.jp/science-engineering/
```

のようにリンク付きで表記される．なお，`url` 命令の中にアドレスを書くときには，中に `_` などの特殊文字があっても問題ない．

5.13 ソースコードの挿入

プログラムを作成するレポートでは，作成したソースコードを示すためにレポートに挿入することがある．5.2 節で述べたように，LaTeX では単独の改行は無視されるので，ソースコードをそのまま挿入しても正しく出力されない．ソースコードを見やすく挿入する方法はいくつかあるが，ここでは `moreverb` パッケージを使った方法を紹介する．

まず，`moreverb` パッケージを使うために，プリアンブルに以下のように `\usepackage` 命令を追記する．

```
\usepackage{moreverb}
```

その上で，以下のように `listing` 環境の中にソースコードを書く．

```
\begin{listing}{1}
/**
 * Hello, World と表示するサンプル
 */
public class HelloWorld {
   public static void main(String[] args) {
      Systen.out.println("Hello, World");
   }
}
\end{listing}
```

そうすると，以下のように各行の冒頭に行番号をつけてソースコードが出力される．行番号が付くとレポート文中でソースコードの内容について言及する際に便利である．

```
1  /**
2   * Hello, World と表示するサンプル
3   */
4  public class HelloWorld {
5     public static void main(String[] args) {
6        Systen.out.println("Hello, World");
7     }
8  }
```

参考文献

[1] 渡辺徹, 好き好き LaTeX 2_ε 初級編, https://mytexpert.osdn.jp/index.php?%B9%A5%A4%AD%B9%A5%A4%ADLaTeX2e/%BD%E9%B5%E9%CA%D4, 2006.

[2]「物理のかぎしっぽ」TeX に関するメモ，http://hooktail.org/computer/index.php?TeX，2021 年 5 月 17 日閲覧.

以上