

第1章 文書の書き方

この章からしっかり文書を作成していきます。この章では、数式、表、図を使わない文書を扱います。まずは普通の文章を打てるようになりましょう！

1.1 基礎知識

ここではかる〜く必要知識を説明しておこうと思います。細かいことは追々説明しますので、かる〜く読み流してもらえれば十分です。

先ほどの例について、以下で説明を加えていこうと思います。

```
\documentclass{jsarticle}
```

```
\begin{document}
```

```
Hello, \TeX!
```

```
\[\int dx = x + C\]
```

```
\end{document}
```

Hello, \TeX !

$$\int dx = x + C$$

まず、コードの書き方についてです。 \TeX の命令は、キーボードの右上と右下にある \backslash で始めます（windowsでは円マーク¥になります）。 \backslash から始めた部分は、何かの処理（例えば \TeX という合成文字を出力するなど）をするためのものであり、そのまま出力されるわけではありません。

次に、ドキュメントクラスについてです。入力之初めに`\documentclass{jsarticle}`とあります。これは、ドキュメント（文書）のクラス（種類）を指定するものです。基本`jsarticle`で大丈夫なので、おまじないみたいなものだと思ってもらって大丈夫です。最初の行に書くようにしてください。

また、`\begin{document}`と`\end{document}`についてですが、これらで挟まれた部分が本文になります。書きたいことはここに書くようにしてください。

最後に、`\documentclass{jsarticle}`と`\begin{document}`の間ですが、このことを「プリアンブル」といいます。今回の例では何も書いてありませんが、ページレイアウトなどのオプションはここで設定することが多いです。

1.2 改行と半角スペースについて

先の例で1行の文書は書けるようになりました。しかし、改行や改段落の仕方はまだ知りません。改行するときは、改行前の文の最後に \backslashbackslash とつければOKです。また、空白行を作ると、それは改段落とみなさ

れて、次の文章の始めが頭下げ（インデント）されます。改段落はしたいけどインデントはしたくないというときは、その段落の前に`\noindent` とつければ OK です。以下に例を示します。

<code>\documentclass{jsarticle}</code>	
 <code>\begin{document}</code>	
あいうえお	あいうえおかきくけこ
かきくけこ <code>\</code>	さしすせそ
さしすせそ	たちつてと
たちつてと <code>\</code>	なにぬねの
<code>\</code>	
なにぬねの	
 <code>\end{document}</code>	

\TeX ファイル中では改行や半角スペース、タブは普通には出力されません。普通に改行をしても出力に反映されないので、`\` という命令で改行するのです。

なお、文章中に命令（`\` で始まるもの）を用いるときは後ろに半角スペースを入れるのが安全です。これ以降、プリアンブルに何も書かないときは、`\documentclass{jsarticle}`、`\begin{document}`、`\end{document}` を省略します。また、`\begin{document}` と `\end{document}` の間を「ドキュメント」と呼ぶことにします。

1.3 コマンドについて

\TeX で作る文書には、地の文と組版命令（コマンド）の両方が含まれています。地の文は他のワープロソフト（word など）と変わらないですが、このコマンドというものは \TeX に特徴的です。コマンドには色々な種類があり、自分で命令を作ることも可能です。例えば、 \LaTeX という文字は `\LaTeX` というコマンドによって出力されます。

先ほど書いたように、コマンドの後ろには半角スペースを入れて使うのが安全です。半角スペースを入れずに打ってしまうと、続く文まで命令の一章とみなされエラーしてしまうので注意してください。 \TeX ではコマンドの後の半角スペースは区切りのために使われ、出力はされません。この仕様により、英語の文章中などに \LaTeX と打ちたければ、コマンドの後に半角スペースを出力するコマンド `_` を使わなければいけません。これを入れずにただスペースを入れただけでは、コマンドの出力と単語の間が詰まってしまうので気を付けて下さい。

さあ `\LaTeX` を使おう！`\`
 Let's enjoy `\LaTeX` \ command ! `\`
 Let's enjoy `\LaTeX` command ! `\`

さあ \LaTeX を使おう！
 Let's enjoy \LaTeX command !
 Let's enjoy \LaTeX command !

1.4 文書の構造

文書には、文書名・著者名・見出し・段落...といった構造があります。この節ではこれらの表示の仕方を扱おうと思います。

1.4.1 タイトル

タイトルには、文書名・著者名・日付が含まれます。これらの出力は以下のようにして行います。

```
\title{タイトル}
\author{著者}
\date{\today}
\maketitle
```

タイトル
著者名
平成 29 年 10 月 14 日

`\title`, `\author`, `\date` の 3 つは、`\maketitle` の前にあれば（プリアンブルでも）大丈夫です。また、特にこの順番である必要もありません。実際にこれらの出力をするのは `\maketitle` であるので、これはちゃんとドキュメント中（`\begin{document}` と `\end{document}` の間）に書いてください。また、何もしなければタイトルは 1 ページ目の上章に出力されますが、タイトルで 1 ページ使いたいのなら、

```
\documentclass[titlepage]{jsarticle}
```

と、ドキュメントクラスにオプションをつけてください。以下でそれぞれの命令について細かく説明します。

まずタイトルについてですが、長くなり改行が必要な場合、途中で `\\` を入れれば改行できます。

また、著者についてですが、複数名いる場合は `\and` で区切ります。`\and` の後に必ず半角スペースを入れてください。

日付についてですが、`{}` の中に手動で打ち込めばその日付が、`\date` を書かなかつたり `{}` の中を `\today` とすればその文書ファイルを \LaTeX が処理した日付が出力されます。また、`\date{}` のように、`{}` の中に何も書かなければ日付の出力はなくなります。不要なときはそうしてください。

1.4.2 本文の構造

タイトルの書き方がわかったので、今度は文書中の章や節などの書き方を見ていきましょう。

```

\part{これは章}
\section{これは節}
\subsection{これは小節}
\subsubsection{小々節}
\paragraph{段落}
\subparagraph{小段落}

```

第 I 章

これは章

1 これは節

1.1 これは小節

1.1.1 小々節

タイトルや節の細かい設定（例えば節番号を §4 にしたいなど）は後々ページレイアウトの章で扱うと思います。

1.5 文字サイズ

文字サイズを変えることを考えます。文字のサイズは、pt（ポイント）という単位を用いて表すことが多いです。標準を 10pt として色々なサイズがあるので、以下に列挙します。

<code>\tiny</code>	5pt	わーい
<code>\scriptsize</code>	7pt	わーい
<code>\footnotesize</code>	8pt	わーい
<code>\small</code>	9pt	わーい
<code>\normalsize</code>	10pt	わーい
<code>\large</code>	12pt	わーい
<code>\Large</code>	14.4pt	わーい
<code>\LARGE</code>	17.28pt	わーい
<code>\huge</code>	20.74pt	わーい
<code>\Huge</code>	24.88pt	わーい

文書全体の文字サイズを変えたいときは、ドキュメントクラスにオプションをつけて指定します。

```
\documentclass[12pt]{jsarticle}
```

1.6 フォント

文字サイズをいじったら今度はフォントをいじりたくなりますね。とりあえず押さえておきたいものだけ下にまとめます。

<code>\textrm{Roman}</code>	Roman	本文（デフォルト）
<code>\textbf{Boldface}</code>	Boldface	太字
<code>\textit{Italic}</code>	<i>Italic</i>	強調（斜体）
<code>\textgt{ゴシック}</code>	ゴシック	見出し

なお、文字サイズは指定したところだけ変更したい場合、その指定する範囲を`{}`で囲みます。そうしないと、それ以降の文字のサイズが変更されたままになってしまうからです。また、これらは併用もできます。

```
{\large 大きい文字} \\
\textgt{太い文字} \\
{\large \textgt{大きい太字}} \\
\textit{Italic 体} \\
\huge こうすると \\
それ以降にも影響が出ます
```

大きい文字

太い文字

大きい太字

Italic 体

こうすると

それ以降にも影響が
出ます

1.7 文字について

今まではアルファベットや平仮名、漢字などの文字を扱ってきました。これらの文字は打った通りに出力されるので、特に困ったことはありませんでした。しかし、 \LaTeX には、そのまま打っても出力できない記号がいくつかあります。また、今までは英語と日本語しか扱っていなかったので問題はありませんでしたが、他の言語（例えばドイツ語やフランス語、スペイン語など）にあってキーボードでは打てない文字はどうしたらよいのでしょうか。この節ではそういった \LaTeX における様々な文字について扱いたいと思います。

1.7.1 そのまま出力されない記号

先ほど説明したように、 \LaTeX にはそのまま打つだけでは出力されない文字がいくつかあります。それは、

```
# $ % & _ { } \ ^ ~ < > /
```

これらの記号です。最後の3つは数式（次章で扱います）中ならそのまま打てますが、地の文では使えません。これらの記号には特殊な役割があり、そのまま打つとその役割を果たそうとしてエラーや出力ミスが起きてしまいます。これらの記号を出力したいときは、記号の前に`\`をつけて、`\#`などとすればOKです。また、半角スペースやタブもそのまま打つだけでは無視されて出力されません。半角スペースを打ちたいときも、同様に`\`をつけて「`_`」のようにすれば出力されます。なお、全角スペースは1つの文字としてみなされているので、出力されます。

1.7.2 そのまま出力する方法

前小節でどの文字が打てないのかはわかりましたので、今度はそれらを文章中に入りたいときにどうしたらいいかを説明しようと思います（もちろんいちいち\をつけても出力はできますが、面倒です）。

打ちたい記号が数文字程度なら、

```
\verb|...|
```

のように、`\verb` という命令の後の `|` の間に打ちたい記号を入れれば OK です（同じ文字で挟めばよいので、`!` や `$` などでも大丈夫です）。

打ちたいものが複数行に渡る場合は、それらを `\begin{verbatim}` と `\end{verbatim}` で囲えば OK です。

<code>\verb #! </code>	<code>#!</code>
<code>\begin{verbatim}</code>	
<code>\documentclass{jsarticle}</code>	<code>\documentclass{jsarticle}</code>
Hello <code>\TeX</code> !	Hello <code>\TeX</code> !
<code>\end{verbatim}</code>	

なお、半角スペースを示す「`_`」を出力したいときは、`\verb*` や `\begin{verbatim*}`, `\end{verbatim*}` のように、「`*`」をつけてください。

1.7.3 特殊文字

\TeX では様々な特殊文字をコマンドによって出力できるようにしています。例えば \copyright や \S , \pounds や \pounds などです。ただ、あんまり使わないという人が多いと思うのでとりあえず省略します。普通にググってください。まとめてあるサイトがあります。

1.8 コメント

文書を作成していると、入力した文の中で無視してほしい部分が出てきます。例えば書きかけで出力してほしくない部分とか、その部分を書いた日付とか。 \TeX では、`%` を書いた行の `%` 以降の部分は無視され、出力されません。この部分をコメント（注釈）といいます。プログラミングでもよく使いますね。これを書くと改行も無視されますので、それも頭に入れて使ってください。必要に応じてコメントを入れて、編集し直ししやすいように文書を作るのをおすすめします。

あいうえお^^I^^I%ここはコメント かきくけこ	
%2017/5/14 たちつと	あいうえお かきくけこ たちつと
%なにぬねの	

また、例えばよくわからないエラーが出るなどの理由で複数行をコメントアウトしたいときもあるかと思いますが、そのときは、`comment` パッケージの `comment` 環境を使ってみましょう。まずはプリアンブル (`\documentclass` と `\begin{document}` の間) に

```
\usepackage{comment}
```

と書いてください。そして、コメントアウトしたい部分を `comment` 環境に入れて

<pre> ここは出力される。\\ ここも。 \begin{comment} ここは出力されない。 ここも。 なんでも書ける \aaaaaaaaaaaaa エラーもしない。 \end{comment} </pre>	<pre> ここは出力される。 ここも。 </pre>
---	-----------------------------

のようにすれば OK です。ただ、`\end{comment}` の直後に `%` を入れるとエラーするようなので、そこだけは気を付けて下さい（また、`listings` 系とも相性が悪いようです）。

1.9 脚注・欄外への書き込み

文書を書いていたら、(文書中に) 注釈を入れたいときもありますよね。この節では脚注と欄外への書き込みの方法を説明します。

まず、脚注ですが、`\footnote{...}` というコマンドを使って、このように¹ つけたいところの直後に書きます (このように `\footnote{わーい}`)。 `\footnote{...}` の後の半角スペースを忘れないようにしてください (もしくは `{}` でくくってください)。

次に欄外の書き込みについてですが、これは `\marginpar{欄外だよ〜}` のように書けば出力できます (ちょっとエラーしてるので省いておきます)。しかし、ドキュメントクラスが `jsarticle` だと欄外部分はあまり広くないので、これを使いたいならページレイアウトをいじるか、ドキュメントクラスを変えるかしてください (どちらも後の章で説明します)。

1.10 長さの単位

\TeX で使える長さの単位は色々ありますが、あまり使わないものは省いてよく使うものだけ列挙しておこうと思います。換算値も一応載せておきますね。

¹わーい

cm	センチメートル	$1\text{ cm} = 10\text{ mm} = 0.3937\text{ in} = 28.452699\text{ pt}$
mm	ミリメートル	
in	インチ	$1\text{ in} = 2.54\text{ cm}$
pt	ポイント	$1\text{ pt} = \frac{1}{72.27}\text{ in} = 0.03515898\text{ cm}$
zw	全角 width	$1\text{ zw} = 9.2469\text{ pt} = 0.32499\text{ cm}$
Q	級	$1\text{ Q} = 0.25\text{ mm}$
H	齒	$1\text{ H} = 0.25\text{ mm}$

また、他に、長さを表すコマンドがいくつかあるのでそれもまとめておきます。真ん中の列に書いているのはデフォルト値です。なお、これらの値は変更可能です。変更の仕方については後々ページレイアウトの章で扱おうと思います。

<code>\baselineskip</code>	12 pt	1 行の縦の長さ
<code>\hspace</code>	$453\text{ pt} = 49\text{ zw}$	1 行の横の長さ

ちょっとした補足を加えます。読み飛ばしてもらっても構いません。

まず、pt（ポイント）についてですが、Word などでは $1/72$ インチになっているので少しだけサイズが違います。このポイントという単位は、文字の大きさを表す単位として広く使われており、10pt が標準となっています。

また、zw についてですが、これは pTeX と upTeX でしか使えません（初期設定で pdfLaTeX になっていると思うので何もいじっていなければ使えます）。また、上の値は jsarticle でのものであり、jarticle では 9.62216 pt となります。

Q と H についてですが、これらは和製の書籍の組版でよく使われる単位です。文字 14Q，行送り 30H というのが大体の標準らしいです。

1.11 空白などの調整

TeX において入力の際の空白や空白行は無視されてしまいます。しかし、全角スペース（そのまま出力される）を必要数打ち込むのは美しくありません。この節では空白の入れ方を説明します。

なお、下で長さとした部分は、前節で挙げた単位の長さか長さを表すコマンドを入力してください。

<code>_</code>	半角スペース
<code>~</code>	改行されないスペース
<code>\quad</code>	本文の文字と同じ大きさのスペース (本文 10pt なら 10pt)
<code>\qquad</code>	<code>\quad</code> の 2 倍
<code>\,</code>	<code>\quad</code> の $3/18$ ほど
<code>\></code>	<code>\quad</code> の $4/18$ ほど (数式モード内のみ)
<code>\;</code>	<code>\quad</code> の $5/18$ ほど (数式モード内のみ)
<code>\!</code>	<code>\quad</code> の $-3/18$ ほど (数式モード内のみ)
<code>\hspace{長さ}</code>	行頭・行末では出力されない水平方向の空白
<code>\hspace*{長さ}</code>	行頭・行末でも出力される水平方向の空白
<code>\vspace{長さ}</code>	行頭・行末では出力されない垂直方向の空白
<code>\vspace*{長さ}</code>	行頭・行末でも出力される垂直方向の空白
<code>\hfill</code>	いくらでも伸びる空白 (弱め)
<code>\hfill</code>	いくらでも伸びる空白 (強め)

適宜これらのスペースをいれて空白の量を調整してください。

また、改行や改段落、改ページの仕方についてですが、一般に改行は`\\`、改段落は空行で行うことが多いです。ただ、明示的にコマンドで行いたいときは、以下の表のようにしてください。なお、インデントとは、段落の最初の頭下げのことです。

<code>\\</code>	改行
<code>\newline</code>	改行
<code>*</code>	改行 (改行位置で改ページしない)
<code>\newline*</code>	改行 (改行位置で改ページしない)
空行	改段落 (インデントあり)
<code>\par</code>	改段落 (インデントあり)
<code>\newpage</code>	改ページ
<code>\clearpage</code>	改ページ (図と表を出力してから)
<code>\samepage</code>	改ページしない

特に、環境 (後で説明します) の後の空白行で`\\`を使うとエラーすることがありますので、そのときは`\vspace*`等を使ってください。

また、この「長さ」にはマイナスの値を指定することもできるので、スペースが空きすぎていると思ったらマイナス値を指定してください。

1.12 罫線と枠

下線や枠線などの引き方を列挙します。

- 下線

`\underline{...}` で下線が引けます。例えばこんな風に (`\underline{例えばこんな風に}`)

- 罫線

`\hrulefill` で罫線が引けます。

- 点線
`\dotfill` で罫線が引けます。
.....
- 枠
`\fbox{...}` で枠が描けます。例えば `\fbox{こんな風に}`
- 幅指定した枠
`\framebox[幅][位置]{...}` で幅と文字の位置を指定した枠を描けます。位置は, l,c,r (left, center, right) の頭文字) から選べます。位置を指定しなければデフォルトの center になります。

わーい
- 長方形
`\rule[d]{w}{h}` でベースラインからの高さ d , 幅 w , 縦の長さ h の長方形が描けます。 d, w, h を調整することで様々な罫線が引けます。この長方形は幅 $w = 0$ にして挿入し、行送りを調整するのによく使われます。

1.13 ルビ

\TeX には様々なパッケージというものがあります。パッケージとは、様々なマクロ（コマンドの定義）をまとめて書いてあるファイルのことです。これを読み込むことで、 \TeX のコマンドを拡張することができます。この節で扱うルビや、次章で扱う数式を（拡張して）使うときにもパッケージを読み込みます。

ルビを振るために読み込むパッケージは、「pxrubrica」です。これを読み込むには、プリアンブルに

```
\usepackage{pxrubrica}
```

と書けば OK です。

これで準備は終わりました。ここから具体的に使い方を見ていきます。ルビの振り方は、

```
\ruby{親文字}{ルビ文字}
```

です。ここで、ルビ文字には、

```
\ruby{漢字}{かん|じ}
```

のように、親文字の区切りの部分に | を入れます（親文字が 1 文字ならいりません）。

また、漢字全体として読みが与えられている場合は、次のようにオプションに「g」を指定します。

```
\ruby[g]{雲雀}{ひばり}
```

以下に例を示します。

<code>\ruby{漢字}{かん じ} \\\</code>	かん じ 漢字
<code>\ruby{獺}{かわうそ} \\\</code>	かわうそ 獺
<code>\ruby{獺祭}{だっ さい} \\\</code>	だっさい 獺祭
<code>\ruby{獺祭}{かわうそ まつり} \\\</code>	かわうそまつり 獺 祭
<code>\ruby{航空宇宙}{こう くう う ちゅう} \\\</code>	こうくううちゅう 航空宇宙
<code>\ruby[g]{雲雀}{ひばり} \\\</code>	ひばり 雲雀
<code>\ruby[g]{超電磁砲}{レー ル ガン}</code>	レー ル ガン 超電磁砲

これで出力されるのは、デフォルトで設定された正しい日本語のルビの振り方によるルビです。ですが、出力が気に入らなかったときは、様々なオプションでその設定を変えることができます。それについては（不要だと思うので）ここでは扱わないので、必要を感じた人はググってみてください。

また、プログラム言語の `ruby` を読み込むパッケージを使っている場合、`\ruby` という命令はそちらの出力をするものとなってしまいます。この場合を考慮して、ルビを振る命令には、`\jruby` という別名も用意されているので、`ruby` を使っている場合はそちらで対応してください。

なお、この節は「[LaTeX 文書で“美しい日本の”ルビを使う ～pxrubrica パッケージ～](#)」を参考にしました。

1.14 圏点

圏点とは文字の上にある強調を示す点のことです。`\.{強}``\.{調}` のようにすれば強調出来ます。もっと点を大きくしたいときは、ルビを使って

```
\ruby{圏点}{\textbullet|\textbullet}
```

のようにすれば強調 出来ます。

ルビを使わない場合は、マクロを扱う章で扱います（そのうち書きます...）。

以上で数式・表・図などを扱わない平の文書を、文字サイズやフォントの変更も含めて書けるようになりました。次章から平ではない部分の書き方を説明していこうと思います。