

論文・レポート執筆のための L^AT_EX 入門会
～環境構築から BibTeX まで～

大芝 峻平

2024 年 10 月 23 日

目次

第 1 章

導入

1.1 $\text{T}_{\text{E}}\text{X}(\text{L}\text{A}\text{T}_{\text{E}}\text{X})$ とは

$\text{T}_{\text{E}}\text{X}$ 「テフ, テック」は, Donald Ervin Knuth 氏 (以下, Knuth 氏) が製作した組版システム [?] で, 現在はそれを基にした様々なバージョンが存在する. また, $\text{L}\text{A}\text{T}_{\text{E}}\text{X}$ 「ラテフ, ラテック」は $\text{T}_{\text{E}}\text{X}$ を基に, マクロパッケージが組み込まれた組版処理システムで, 高品質かつ自由度の高い組版処理能力と, マクロパッケージに由来する扱いやすさを特徴とする.

1.2 組版とは

組版とは, 原稿及びレイアウト (デザイン) の指定に従って, 文字・図版・写真などを配置する作業の総称.

1.3 $\text{L}\text{A}\text{T}_{\text{E}}\text{X}$ の特徴・利点

$\text{L}\text{A}\text{T}_{\text{E}}\text{X}$ の特徴として, 先述した通り卓越した組版処理能力, 扱いやすさはもちろんのこと, 特筆すべきは章番号, 図表番号が自動で振られること, そして数式のデザインが Microsoft Word よりも多彩であることである. 後述するコマンドを上手く使えば, あなたが望むままにレポートを作成することが可能であろう.

1.4 コンパイラ

$\text{T}_{\text{E}}\text{X}$ では C 言語のようにコードから PDF に書き出す際に**コンパイラ**によって変換される. C 言語でも gcc, Visual C++ とコンパイラに様々な種類があるように, $\text{T}_{\text{E}}\text{X}$ でも pLaTeX や LuaLaTeX, XeLaTeX のように様々なコンパイラが存在する. 本書ではフォント等の自由度が高い LuaLaTeX でのコンパイルを前提として説明する. 基本的には文法は大きく変わらない為, 高速な pLaTeX でのコンパイラも各自で試してみたい.

1.5 プリアンブル

L^AT_EX で文書を作成する際、本文の前に書かれる部分を**プリアンブル**と呼ぶ。プリアンブルには、様々な設定を記述することができる。例えば、用紙の大きさやフォントの設定、パッケージの読み込みなどがある。

1.6 クラス

L^AT_EX ではクラスと呼ばれる概念がある。これは、基本的には何を作るのかによって変わる。例えば、レポートや論文を書くのであれば `article` クラスを用いる。ただ、`article` クラスで記述できるのは英語論文であるため、日本語で論文を書きたいならば `jsarticle` を用いなければならない。また、`luatex` をコンパイラに用いる場合には `ltjsarticle` クラスが必要になり、細かい仕様が変わってくる。他にも、本を書くための `book` クラスがあったり、プレゼン資料を書くための `Beamer` クラスがあったり、更に学会によっては論文のテンプレートに独自のクラスファイルが用いられていることが多い。本書では LuaLaTeX を用いるので、記載されているコードは少なくとも `ltjsarticle` クラスで動作する。基本的には pLaTeX でも動くようなコードが中心だが、一部動かないコードが存在するので注意が必要。実際には以下のように用いる。

```
1 \documentclass[a4paper, titlepage, 10pt]{ltjsarticle}
```

`a4paper` は A4 用紙を指定するオプションで、他にも `a5paper` などが指定できる。また、`titlepage` はタイトルページを作るオプションで、指定しなければ作られない。`10pt` は文字の基本サイズを表すオプションで、`9pt`~`12pt` くらいをよく指定する。

1.7 パッケージ

L^AT_EX は、素の状態で作成しようとしても、自分が作りたいように作成するのは非常に難しい。そこで、L^AT_EX の機能を拡張するための**パッケージ**が用意されている。目的や用途によってパッケージが分かれており、自分の目的に合ったパッケージを読み込むことで、より効率的、かつ綺麗に文書を作成することができる。使用方法など、詳細は後述する。

第 2 章

環境構築

2.1 ローカルでの環境構築

TeX でコンパイルを行うためには, TeX Live をインストールする必要がある. 本書執筆時点 (2024 年 10 月 23 日現在) では, TeX Live 2024 が最新バージョンであるため, ここでは, TeX Live 2024 のインストール方法を紹介する.

2.2 TeX Live 2024 のインストール

2.2.1 Windows の場合

`https://mirror.ctan.org/systems/texlive/tlnet/install-tl-windows.exe` からインストールファイルをダウンロードする. ダウンロードが終わったら, エクスプローラを開き, `install-tl-windows.exe` を起動する. このとき, ファイル名を **右クリック**して「管理者として実行」をクリックすると, 全てのユーザ向けにインストールすることができるため, 必要に応じて管理者権限で実行すると良い.

exe ファイルを実行すると, 次のようなウィンドウが立ち上がる. デフォルトで `install` が選ばれているので, `install` にチェックを入れたまま「Next >」をクリックする. 次のウィンドウでもそのまま「Install」を押せば, インストールが開始される. なお, この作業は非常に時間がかかるため, 注意が必要.

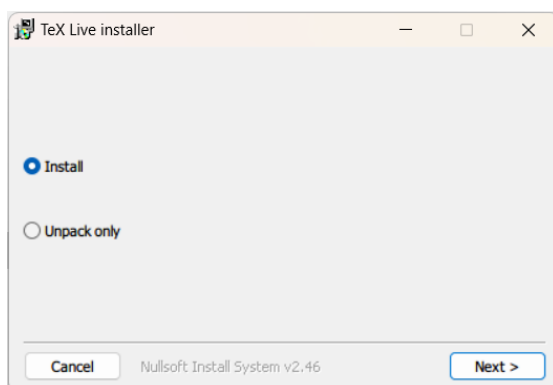


図 1: インストールウィンドウ (Windows)(1)

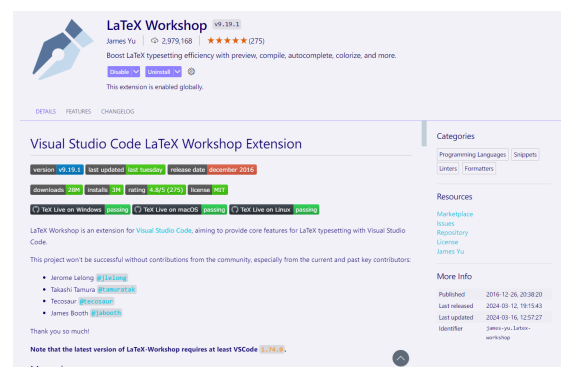


図 2: LaTeX Workshop 拡張機能

2.2.2 Linux (Ubuntu) の場合

Linux では、流れとしては Windows でのインストール方法と大差は無いが、基本的にコンソール上ですべての工程を行う。まず、ミラーサイトから `install-tl-unx.tar.gz` をダウンロードする必要があるので、`wget` または `curl` コマンドを使用する。

`wget` コマンドの場合は、

```
wget http://mirror.ctan.org/systems/texlive/tlnet/install-  
-tl-unx.tar.gz
```

`curl` コマンドの場合は、

```
curl -OL http://mirror.ctan.org/systems/texlive/tlnet/install-tl-  
unx.tar.gz
```

このコマンドを実行したら、次はダウンロードしたインストーラのファイルを展開する。

```
tar xvf install-unx.tar.gz
```

展開したインストーラのディレクトリに移動する。

```
cd install-tl-2*
```

root 権限でインストーラを実行する。

```
sudo ./install-tl -no-gui -repository http://mirror.ctan.org/syste  
ms/texlive/tlnet/
```

この時、以下のような表示が出るので、`I` を入力してインストールを開始する。

Actions:

```
<I> start installation to hard disk
```

```
<H> help
```

```
<Q> quit
```

Enter command:

インストールが終了したら `/usr/local/bin` ディレクトリは以下にシンボリックリンクを追加する。

```
sudo /usr/local/texlive/????/bin/*/tlmgr path add
```

途中の `?や*` はワイルドカード検索のため、自動的にうまく実行されるはずだが、そうでない場合は以下のように具体的なディレクトリ名を指定する。 `sudo /usr/local/texlive/2024/bin/x86_64-linux/tlmgr path add`

もし以上の解説でうまくいかない場合は、TeXWiki のインストールガイド (<https://texwiki.texjp.org/?Linux>) を参照してほしい。

2.2.3 Mac OS の場合

Mac OS では、Mac 向けの TeX Live のパッケージである MacTeX の導入が推奨されている。基本的にはフルインストールを推奨するので、以下にフルインストールのためのコマンドを紹介する。

- GUI アプリケーションありの場合

```
brew install --cask mactex
sudo tlmgr update --self --all
sudo tlmgr paper a4
```
- GUI アプリケーションなしの場合

```
brew install --cask mactex-no-gui
sudo tlmgr update --self --all
sudo tlmgr paper a4
```

Homebrew が入っていない場合は Homebrew 公式 HP からダウンロード・インストールすること。

Homebrew 日本語公式ホームページ: <https://brew.sh/ja/>

2.3 VS Code に TeX の拡張機能を追加する

TeX Live のインストールが終われば、次は VS Code から $\text{T}_{\text{E}}\text{X}$ をコンパイルできるようにする必要がある。まず、 $\text{T}_{\text{E}}\text{X}$ の拡張機能をインストールしよう。VS Code の「拡張機能」にて、「LaTeX Workshop」と検索すれば同名の拡張機能が出てくるため、それをインストールする。(図??参照)

基本的には以上で作業は完了である。

第 3 章

L^AT_EX の基本・テキスト編

3.1 パッケージ

3.1.1 パッケージとは・パッケージの使い方

L^AT_EX の最大の特徴はパッケージによって多彩な機能を追加できる点である。ではパッケージとはどういうものかという、いくつかの機能をまとめて使えるようにした、いわばお道具箱のようなものである。例えば、テキストに色をつけたい時、`xcolor` パッケージを読み込めばそれを使うことができる。どのように読み込めば良いかという。

```
\usepackage{xcolor}
```

とプリアンブルに書くことで読み込むことができる。

3.2 タイトルの表示

3.2.1 タイトル、著者、執筆日の設定

L^AT_EX で文書を作成する際には、タイトル、著者、執筆日を設定する必要がある。タイトルを設定するには、`title` コマンドを使用する。著者を設定するには、`author` コマンドを使用する。執筆日を設定するには、`date` コマンドを使用する。

```
1 \title{タイトル}
2 \author{著者}
3 \date{執筆日}
```

3.2.2 タイトルの表示

タイトル、著者、執筆日を設定した後、`document` 環境内で `maketitle` コマンドを使用することで、それらを表示することができる。

```
1 \begin{document}
2   \maketitle
3 \end{document}
```

3.2.3 document 環境とは

document 環境は, begin コマンドと end コマンドで囲まれた部分を指す. document 環境内に本文を記述する. ここに記述された内容が実際に文書として表示される. 逆に, document 環境外に記述された内容は, 文書として表示されずにエラーが起きる.

3.3 目次

3.3.1 目次の表示

目次を表示するには, tableofcontents コマンドを使用する. clearpage コマンドを合わせて使用することで, 目次を新しいページに表示することができる.

```
1 \tableofcontents
2 \clearpage
```

3.4 章立て

3.4.1 章立ての方法

レポートにおいて, 章立ては必須である. 章立てをする際には以下のタグを用いる.

- \section
- \subsection
- \subsubsection
- \paragraph

section は行った実験ごとに章を分ける場合に使用し, subsection はその実験の各項目 (目的, 実験方法など) を分けるのに使用する場合が多い. subsubsection に関しては更に細かく章を分けたいときに使用する. paragraph は, 更に細かい章分けに用いる.

例えば、工学部 2 年後期から始まる実験では、数日に分けて実験を行う場合が多いので、以下のように章立てをするのが良いだろう。

—— 章立ての例 ——

```
\section{1 日目 実験内容}
\subsection{実験目的}
:
\subsection{考察}
\section{2 日目 実験内容}
\subsection{実験目的}
:
```

学部 4 年生以上は、所属する研究室や論文を提出する学会のルールに従うこと。

3.5 文字の装飾

論文やレポートなどを執筆したいとき、**太字**や *Italic*、**色付き文字**などを使って強調したいことがあるだろう。そこで L^AT_EX で使える文字のスタイルズコマンドを以下に示す。

3.5.1 太字

太字を挿入したいときは、`textbf` コマンドを使用する。具体的には、次のように使う。

—— 太字の例 ——

```
\textbf{太字にしたい文}
```

3.5.2 Italic(斜体)

斜体に関しては、日本語フォントに斜体が組み込まれていないため、基本的には日本語の斜体はサポートされていない。正確には全くできないというわけではないが、複雑かつ体裁が崩れやすいため、本誌では紹介しない。英語に関してはシンプルな手法でできるため、以下に斜体にするためのコマンドを示す。

—— 斜体の例 ——

```
\textit{斜体にしたい英文}
```

3.5.3 等幅

ソースコードを一部示すときなど、一時的に等幅フォントを使用したい場合は `texttt` コマンドを使用する。具体的には、次のように使う。

—— 等幅の例 ——

```
\texttt{等幅にしたい文}
```

3.5.4 色付き文字

テキストの一部に色をつけたい場合は、`color` コマンドを使用する。なお、使用できる色については読み込むパッケージに依存しており、`xcolor` パッケージではさまざまな色が使える。逆にパッケージを読み込まなければ多彩な色付き文字を使うことはできないので、冒頭で `xcolor` パッケージを読み込まなければならない。具体的には、次のように使う。

—— 色付き文字の例 ——

```
\usepackage[dvipsnames]{xcolor} % xcolor パッケージを dvipsnames で  
読み込み（これで多彩な色使える）
```

中略

```
\color{色}色付きにしたい文\color{black}
```

`xcolor` パッケージで利用できる色については、OverLeaf のドキュメントを参考にすると良い。 (https://ja.overleaf.com/learn/latex/Using_colors_in_LaTeX)

第 4 章

さまざまな「環境」

4.1 「環境」とは

図表を挿入する際、 \LaTeX では「環境 (environment)」というものを宣言し、その中に図表を挿入する。環境は挿入するものによって分けられており、以下のように対応している。

箇条書き : `itemize` 環境 (順番をつけるときは `enumerate` など)

数式 : `align` 環境, `equation` 環境など

図 : `figure` 環境

表 : `table` 環境

他にも様々な環境が存在するが、本誌では代表的な環境を紹介する。また、環境を使用するときは基本的に `begin` コマンドではじめ、`end` コマンドで終了する。例えば、`itemize` 環境を用いるときは以下のようなになる。

環境の使い方

```
\begin{itemize}
  \item アイテム A
  \item アイテム B
\end{itemize}
```

4.2 箇条書き

通常の箇条書きには `itemize` を用いる。また、箇条書きする内容の先頭には `\item` とつける必要がある。他にも箇条書きで用語を説明する `description` や、数字がつく `enumerate` も存在する。それぞれの例を以下に示していく。

4.2.1 itemize の場合

itemize 環境を使う時のソースコード

```
\begin{itemize}
  \item アイテム A
  \item アイテム B
\end{itemize}
```

- アイテム A
- アイテム B

4.2.2 description の場合

description 環境を使う時のソースコード

```
\begin{description}
  \item[説明 A] アイテム A
  \item[説明 B] アイテム B
\end{description}
```

説明 A アイテム A

説明 B アイテム B

4.2.3 enumerate の場合

enumerate 環境を使う時のソースコード

```
\begin{enumerate}
  \item アイテム A
  \item アイテム B
\end{enumerate}
```

1. アイテム A
2. アイテム B

4.3 数式

一行で完結する数式，または複数行の数式に一つの式番号を振りたい場合は `equation` を，複数行の数式にそれぞれ連続した式番号を振りたい場合は `align`

を用いる。なお、式番号を振りたくない場合は「*(アスタリスク)」を `equation` や `align` の直後につける。

4.3.1 equation

— equation —

```
\begin{equation}
  e^{i\pi}=-1
\end{equation}
\begin{equation}
  \begin{split}
    \cos^2\theta &= \cos^2\theta - \sin^2\theta \\
    &= 2\cos^2\theta - 1 \\
    &= 1 - 2\sin^2\theta
  \end{split}
\end{equation}
```

$$e^{i\pi} = -1 \tag{1}$$

$$\begin{aligned} \cos 2\theta &= \cos^2 \theta - \sin^2 \theta \\ &= 2 \cos^2 \theta - 1 \\ &= 1 - 2 \sin^2 \theta \end{aligned} \tag{2}$$

4.3.2 align

— align —

```
\begin{align}
  \cos^2\theta &= \cos^2\theta - \sin^2\theta \\
  &= 2\cos^2\theta - 1 \\
  &= 1 - 2\sin^2\theta
\end{align}
```

$$\cos 2\theta = \cos^2 \theta - \sin^2 \theta \tag{3}$$

$$= 2 \cos^2 \theta - 1 \tag{4}$$

$$= 1 - 2 \sin^2 \theta \tag{5}$$

4.4 図

図を挿入する際には `figure` 環境を用いる。 `figure` 環境を用いることで、図にキャプションをつけることができる。また、図の位置を指定することもできる。

— `figure` 環境を使う時のソースコード —

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=12cm]{sample.jpg}
  \caption{画像サンプル}
\end{figure}
```



図 3: 画像サンプル

4.5 表

表を挿入する際には `table` 環境を用いる。 `table` 環境を用いることで、表にキャプションをつけることができる。また、表の位置を指定することもできる。

table 環境を使う時のソースコード

```
\begin{table}[htbp]
  \centering
  \caption{表サンプル}
  \begin{tabular}{c|c}
    \hline タイトル A & タイトル B \\
    \hline 内容 C & 内容 D \\
    \hline 内容 E & 内容 F \\
    \hline
  \end{tabular}
\end{table}
```

表 1: 表サンプル

タイトル A	タイトル B
内容 C	内容 D
内容 E	内容 F

4.6 htbp とは

図や表の位置を指定する際に、htbp というオプションを指定することがある。htbp はそれぞれ以下のような意味を持つ。

h : その場所に挿入

t : ページの上部に挿入

b : ページの下部に挿入

p : 1 ページにまとめて挿入

何も指定しない場合は、 \LaTeX が自動で最適な位置に挿入する。しかし、ページ数が一番少なくなるような位置に挿入するため、図や表が思った位置に挿入されないことが多い。また、htbp 指定でも必ずしも思った位置に挿入されるわけではない。どうしても図や表を特定の位置に挿入したい場合は、here パッケージを使って [H] オプションを指定することで、その場所に図表を挿入できる。

第 5 章

参考文献の挿入

参考文献を挿入するときは BibTeX を用いるのが便利である。BibTeX を用いると、文献情報を記述したファイル (.bib) を作成し、それを参照することで、文献リストを自動で生成することができる。

5.1 BibTeX の使い方

BibTeX を用いるためには、以下の手順を踏む必要がある。

1. 参考文献情報を記述したファイル (.bib) を作成する
2. L^AT_EX ファイル内で BibTeX を読み込む
3. 参考文献リストを挿入する

5.2 参考文献情報の記述

参考文献情報を記述したファイル (.bib) は、以下のような形式で記述する。

5.2.1 書籍の場合

```
1 @book{book1,  
2   author = "著者名",  
3   title  = "書籍名",  
4   publisher = "出版社",  
5   year   = "出版年",  
6 }
```

5.2.2 論文の場合

```
1 @article{article1,  
2   author = "著者名",  
3   title  = "論文名",  
4   journal = "雑誌名",  
5   volume = "巻数",  
6   number = "号数",  
7   pages  = "ページ数",
```

```
8   year = "発行年",  
9 }
```

5.2.3 ウェブサイトの場合

```
1 @misc{misc1,  
2   author = "著者名",  
3   title = "ページタイトル",  
4   howpublished = "URL",  
5   year = "アクセス年月日",  
6 }
```

なお、書籍や論文の場合は、必要に応じて `volume`, `number`, `pages` を記述する。

5.3 BibTeX ファイルの読み込み

BibTeX ファイルを読み込むためには、以下のコマンドを \LaTeX ファイル内に記述する。

```
1 \bibliography{参考文献ファイル名}
```

5.4 参考文献リストの挿入

参考文献リストを挿入するためには、以下のコマンドを \LaTeX ファイル内に記述する。

```
1 \bibliographystyle{スタイル名}  
2 \bibliography{参考文献ファイル名}
```

スタイル名には、`jplain`, `junsrt` などがある。`jplain` は参考文献リストをアルファベット順に並べるスタイルである。一方、`junsrt` は参考文献リストを引用順に並べるスタイルである。

第 6 章

体裁の調整

6.1 ページレイアウト

余白を調整する方法には、いくつかの方法がある。ここでは、geometry パッケージを用いた方法とパッケージを用いない方法を紹介する。

6.1.1 geometry パッケージを用いた方法

geometry パッケージを用いると、簡単にページレイアウトを調整することができる。geometry パッケージを用いると、以下のように記述することで、ページレイアウトを調整することができる。

```
1 \usepackage{geometry}
2 \geometry{left=3cm,right=3cm,top=3cm,bottom=3cm}
```

この場合、左右上下の余白をそれぞれ 3cm に設定している。また、このように指定する方法もある。

```
1 \usepackage[top=3cm,bottom=3cm,left=3cm,right=3cm]{geometry}
```

6.1.2 パッケージを用いない方法

パッケージを用いない方法でページレイアウトを調整する場合は、以下のように記述する。

```
1 % 余白の設定 ---
2 \setlength{\textheight}{\paperheight}
3 \setlength{\topmargin}{4.6truecm} % 上の余白を30mm (=1inch[25.4mm] - 4.6mm
   %)に
4 \addtolength{\topmargin}{-\headheight}
5 \addtolength{\topmargin}{-\headsep}
6 \addtolength{\textheight}{-60truecm} % 下の余白も30mmに (TOP+20mm)
7
8 \setlength{\textwidth}{\paperwidth}
9 \setlength{\oddsidemargin}{4.6truecm} % 左の余白を30mm (=1inch[25.4mm] +
   4.6mm)に
10 \setlength{\evensidemargin}{\oddsidemargin}
```

```

11 \addtolength{\textwidth}{-61trueem} % 右の余白も30mmに
12 % ---

```

ちなみにこれは本書の設定である。また、インデントなどの幅を細く変えたいとき、単位を用いる時がある。このとき、単位として使用できるのは以下の通りである。

- mm - ミリメートル。1mm は 1 ミリメートルに相当する単位。
- cm - センチメートル。1cm は 10 ミリメートルに相当する単位。
- in - インチ。1in は 2.54cm (25.4mm) に相当する単位。
- pt - ポイント。1pt は 1/72 インチ (約 0.3527mm) に相当する単位。
- em - em 単位。その場で使われているフォントサイズ (ポイント数) に相当する相対単位。例えば、12pt のフォントなら 1em は 12pt。
- ex - ex 単位。フォントの x 高さに基づく相対単位で、x 文字の高さを基準とする。フォントによって変動する。
- mu - mu 単位。1mu は 1/18em に相当する非常に小さな単位で、主に数式内の余白調整に使用される。
- zh - 全角文字の高さに基づく単位。CJK (中国語, 日本語, 韓国語) 用の文字サイズに対応。
- zw - 全角文字の幅に基づく単位。
- % - パーセンテージ。例えば、ページやカラムの幅の何 % かを指定する際に使用する。
- \linewidth - 現在の段落の幅を指定する。段落内での相対的な長さ調整に使える。
- \textheight - 本文領域の高さを指定する。ページ内の文章領域のサイズ調整に使われる。
- \paperheight - 紙全体の高さを指定する。文書全体のレイアウト設計に使用する。
- \paperwidth - 紙全体の幅を指定する。文書全体のレイアウト設計に使用する。
- \textwidth - 本文領域の幅を指定する。文章の横幅を設定するための重要な単位。
- \columnwidth - カラムの幅を指定する。2 カラムや複数カラムのレイアウトで使われる。
- \columnsep - カラム間の間隔を指定する。カラム間の余白の調整に使用される。
- \columnseprule - カラム間のルールの幅を指定する。カラム間の野線の太さを調整できる。

6.2 ページスタイル

fancyhdr パッケージを用いると、ページスタイルを少し変えることができる。以下のように記述することで、ページスタイルを変更することができる。

```
1 % ページスタイルの設定 ---
2 \pagestyle{fancy}
3 \lhead{}
4 \chead{}
5 \rhead{}
6 \lfoot{}
7 \cfoot{--\;\thepage\;--}
8 \rfoot{}
```

この場合、ページの中央下部にページ番号が表示される。本書では、各章で右上に章番号と章名、中央下にページ番号を表示するように設定している。

6.3 段組

multicol パッケージを用いると、段組を簡単に設定することができる。以下のように記述することで、段組を設定することができる。

```
1 \usepackage{multicol}
2 \begin{multicols}{2}
3   ここに段組したい文章を記述する.
4 \end{multicols}
```

この場合、2つの段組を設定している。他には、documentclass のオプションで twocolumn を指定することで、2段組にすることもできる。

第 7 章

TikZ

7.1 TikZ とは

TikZ(読み: ティクス, ティクツなど) とは, Beamer の作成者である Till Tantau 氏によって作成された描画パッケージである.

7.2 TikZ の導入方法

TikZ を使うときに LuaLaTeX の良さが現れる. LuaLaTeX では他のパッケージと同様, 以下のように宣言すれば良い.

```
1 \usepackage{tikz}
```

一方, pLaTeX などでは以下のようにパッケージの使用を宣言しなければならない.

```
1 \usepackage[dvipdfmx]{graphicx, xcolor}
2 \usepackage{tikz}
```

これは, pLaTeX では tex ファイルから DVI ファイルという形式で書き出してから, PDF に変換するという都合上, dvipdfmx というドライバを指定する必要があるからだ. したがって, グラフィック関係のパッケージを読み込むときに dvipdfmx ドライバを指定するか, documentclass を書くときに dvipdfmx ドライバをオプションとして指定しなければならない.

7.3 TikZ 用のライブラリ

TikZ には専用のライブラリがあり, それらは `usetikzlibrary` コマンドで呼び出せる. 多数あるため自由度が高いので, 設定例を以下に示す.

```
1 \usepackage{tikz, tikz-3dplot, braket}
2 \usetikzlibrary{patterns, intersections, calc, quotes, angles, arrows.
   meta, through, shapes.geometric}
3 \usetikzlibrary{shapes, positioning}
```

7.4 グラフの描き方

TikZ を使ってグラフを描くときは、`tikzpicture` 環境を使う。

```
1 \begin{tikzpicture}
2   \draw (0,0) -- (1,1);
3 \end{tikzpicture}
```

このコードは、原点から (1,1) までの直線を描く。

7.4.1 矢印の描き方

矢印を描くときは、`->`または`-latex`を使う。

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \draw[->] (0,0) -- (1,1);
4     \draw[-latex] (3,0) -- (4,1);
5   \end{tikzpicture}
6 \end{figure}
```

7.4.2 線の太さの変更

線の太さを変更するときは、`line width`を使う。

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \draw[line width=2pt] (0,0) -- (1,1);
4   \end{tikzpicture}
5 \end{figure}
```

7.4.3 線の色の変更

線の色を変更するときは、`color`を使う。

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \draw[color=red] (0,0) -- (1,1);
4   \end{tikzpicture}
5 \end{figure}
```

7.4.4 線のスタイルの変更

線のスタイルを変更するときは, `dashed` を使う.

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \draw[dashed] (0,0) -- (1,1);
4   \end{tikzpicture}
5 \end{figure}
```

7.4.5 線のパターンの変更

線のパターンを変更するときは, `dotted` を使う.

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \draw[dotted] (0,0) -- (1,1);
4   \end{tikzpicture}
5 \end{figure}
```

7.4.6 線の角の変更

線の角を変更するときは, `rounded corners` を使う.

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \draw[rounded corners] (0,0) -- (1,1);
4   \end{tikzpicture}
5 \end{figure}
```

7.4.7 線の始点と終点の変更

線の始点と終点を変更するときは, `shift` を使う.

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \draw[shift={(1,1)}] (0,0) -- (1,1);
4   \end{tikzpicture}
5 \end{figure}
```

7.4.8 線の透明度の変更

線の透明度を変更するときは, `opacity` を使う.

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \draw[opacity=0.5] (0,0) -- (1,1);
4   \end{tikzpicture}
5 \end{figure}
```

7.4.9 点のプロット

点をプロットするときは, `fill` を使う.

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \fill (0,0) circle [radius=1pt];
4   \end{tikzpicture}
5 \end{figure}
```

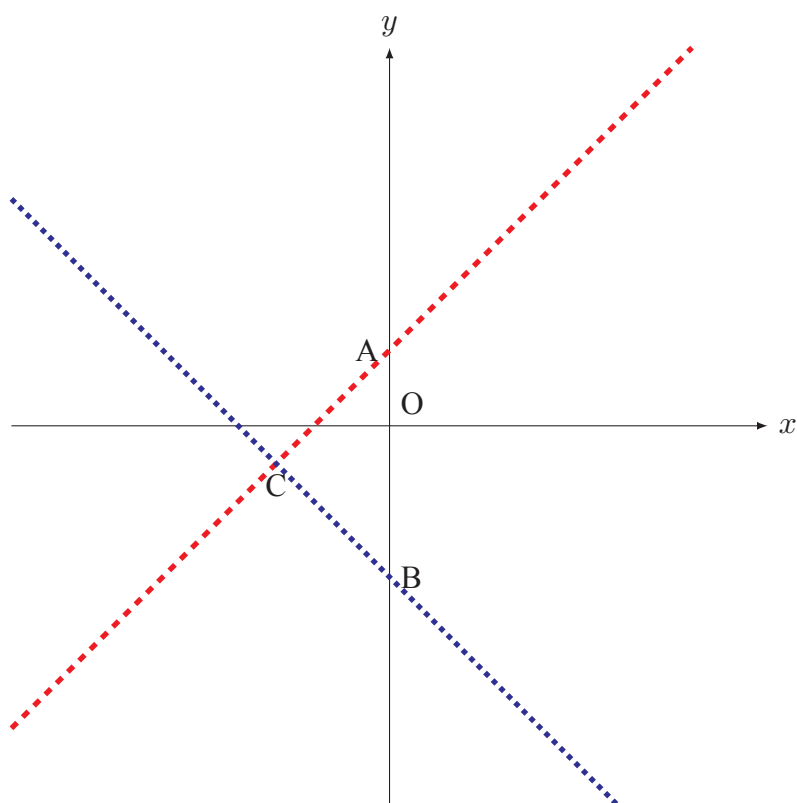
7.4.10 点のラベル

点にラベルをつけるときは, `node` を使う.

```
1 \begin{figure}
2   \begin{tikzpicture}
3     \node at (0, 1) [left] {A};
4   \end{tikzpicture}
5 \end{figure}
```

7.4.11 まとめ

以上のことをまとめると、以下のようなグラフを作成できる。



```

1  \begin{figure}[H]
2    \centering
3    \begin{tikzpicture}
4      \draw[-latex] (-5,0) -- (5,0) node[right] {$x$};
5      \draw[-latex] (0,-5) -- (0,5) node[above] {$y$};
6      \draw[color=red, dashed, line width=2pt] (-5,-4) -- (4,5);
7      \draw[color=blue, dotted, line width=2pt] (-5,3) -- (3,-5);
8      \node at (0, 0) [above right] {O};
9      \node at (0, 1) [left] {A};
10     \node at (0, -2) [right] {B};
11     \node at (-1.5, -0.5) [below] {C};
12   \end{tikzpicture}
13 \end{figure}

```

7.5 TikZ でシステム構成図やフローチャートを作る

前節では、TikZ によるグラフの作成について記載したが、本節では TikZ によるシステム構成図の描画について解説する。

7.5.1 基本構造

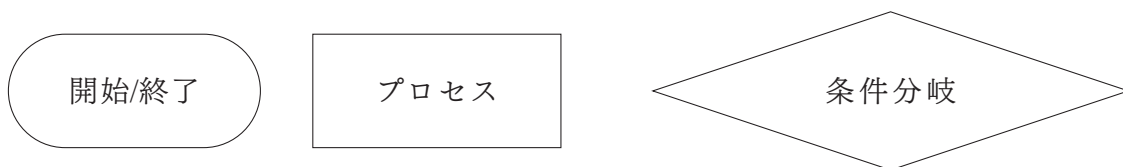
基本的に TikZ でシステム構成図やフローチャートを作成する場合、ほぼ `node` コマンドしか使用しない。また、フローチャートを記述する場合は図形の使い回しが発生するので、ある程度の設定を `tikzset` コマンドでまとめておくといい。以下にフローチャートで主に使う図形の設定をまとめておく。

```

1 \begin{tikzpicture}
2   \tikzset{Terminal/.style={rounded rectangle, draw, text centered, text
      width=3cm, minimum height=1.5cm}};
3   \tikzset{Process/.style={rectangle, draw, text centered, text width=3cm
      , minimum height=1.5cm}};
4   \tikzset{Decision/.style={diamond, draw, text centered, aspect=3, text
      width=5cm, minimum height=1.5cm}};
5 \end{tikzpicture}

```

この設定では以下のような形状のノードが作成できる。



7.5.2 ノードの書き方

まずは前項から出てきた「ノード」について軽く解説する。「結び目」や「集合点」を意味し、例えばフローチャートでは一連の行動の中で「結び目」を作ってそこに行動を記録する、といった感じである。ノード (点) を線で繋いでフローチャートを作っていく、というイメージを持つといい。

さて、実際にどのような記法になるかであるが、例を見た方がわかりやすいだろう。

```
1 \begin{figure}[H]
2   \centering
3   \begin{tikzpicture}
4     % フローチャートで用いる図形の定義 ===
5     \tikzset{Terminal/.style={rounded rectangle, draw, text centered,
6       text width=3cm, minimum height=1.5cm}};
7     \tikzset{Process/.style={rectangle, draw, text centered, text width
8       =3cm, minimum height=1.5cm}};
9     \tikzset{Decision/.style={diamond, draw, text centered, aspect=3,
10       text width=4cm, minimum height=1.5cm}};
11     % ノード ===
12     \node[Terminal] (start){開始};
13     \node[Process, below = 1cm of start] (process1){プロセス1};
14     \node[Decision, below = 1cm of process1] (decision1){条件分岐};
15     \node[Process, below = 1cm of decision1] (process2){プロセス2};
16     \node[Terminal, below = 1cm of process2] (end){終了};
17     % 線 ===
18     \draw[-latex] (start)--(process1);
19     \draw[-latex] (process1)--(decision1);
20     \draw[-latex] (decision1)--node[right]{YES}(process2);
21     \draw[-latex] (decision1.east) |- node[above]{NO}(process1.east);
22     \draw[-latex] (process2)--(end);
23   \end{tikzpicture}
24   \caption{フローチャート}
25 \end{figure}
```

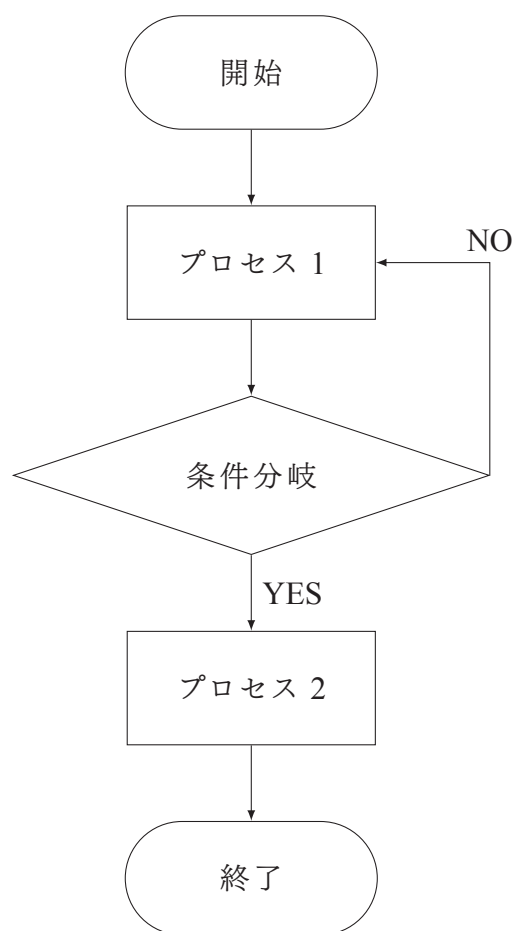


図 4: フローチャート

TikZ のオプションは多彩なので、各自で色々調べつつ上記の例を少しずつ変えてオリジナルのものを作ることをおすすめする。