

# 第1章 図と色

この章では図の挿入と自動配置の仕方、そして色について説明したいと思います。

## 1.1 図

他のソフトを使わず、つまり  $\text{\LaTeX}$  だけでも、例えば `TikZ` というパッケージを使えば図を描くことができます。しかし、これは全てコマンドで描かねばならず、面倒でかつ技術や知識も必要です。

そこで、Excel や PowerPoint、または手書きなどで図を用意することが多いと思います。そうすると、ほとんどの画像が PDF か JPEG で用意されると思うので、それを想定して書いています（おそらく他の形式の図にも通用すると思います）。

### 1.1.1 図の挿入

図の挿入には `graphicx` パッケージを使います。この `graphicx` パッケージは、グラフィック関係全般で使えるパッケージですので、読み込んでおいて損はないです。図を挿入したいときは以下のようにします。

```
\documentclass[dvipdfmx]{jsarticle}
\usepackage{graphicx}
\begin{document}

\includegraphics[オプション]{図のファイル名}

\end{document}
```

`\documentclass` のオプションに `dvipdfmx` というものがあります。この `dvipdfmx` は、ドライバ名で、 $\text{p}\text{\TeX}$ 、 $\text{up}\text{\TeX}$  で標準的に使われるドライバです。PDF、JPEG、PNG、EPS 形式の図を扱うことができます。他に `pdftex`、`xetex`、`luatex` などがありますが、`dvipdfmx` にしておいて大丈夫です。

図の挿入自体には、

```
\includegraphics[オプション]{図のファイル名}
```

というコマンドを使います。

ファイル名には全角文字は使わず、半角アルファベットと数字とアンダーバーだけにした方が安全です。一応、`grffile` というパッケージを使えば大丈夫みたいですけど。

図のファイルは文書と同じところにおけば大丈夫ですが、図だけ別で管理したいという場合、

```
\graphicspath{{フォルダ1のパス}{フォルダ2のパス}...}
```

とすれば、フォルダ1、フォルダ2... の中も探してくれます。また、個々のコマンドでファイル名を絶対パスや相対パスで指定することもできます。パスの話が分からない人は第??章の第??節を読んでください。

`\includegraphics` につけるオプションには以下のようなものがあります。

<code>width=長さ</code>	幅を指定した長さにします。
<code>height=長さ</code>	高さを指定した長さにします。
<code>totalheight=長さ</code>	高さ+深さが指定できます。図を回転したときに便利です。
<code>keepaspectratio</code>	<code>width</code> と <code>height</code> を両方指定したとき、縦横比を変えずに指定した幅、高さに図を入れます。
<code>scale=倍率</code>	画像のサイズが倍率倍されます。
<code>clip</code>	描画領域の外側を描かないという指定です。周囲の文書が侵食されるのを防ぐために指定します。これは指定しておくのが安全です。
<code>trim = x<sub>1</sub> y<sub>1</sub> x<sub>2</sub> y<sub>2</sub></code>	左 $x_1$ , 下 $y_1$ , 右 $x_2$ , 上 $y_2$ だけ図を切り詰めます。単位は 1/72 インチです。
<code>viewpoint= x<sub>1</sub> y<sub>1</sub> x<sub>2</sub> y<sub>2</sub></code>	元の画像の左下の角を原点として、左下の角を $(x_1, y_1)$ , 右上の角を $(x_2, y_2)$ とした長方形の領域を切り出します。こちらも単位は 1/72 インチです。
<code>angle=角度 (単位は度)</code>	画像を指定した角度だけ回転します。
<code>origin=位置</code>	画像の回転の中心を指定します。この位置には、 <code>lrctbB</code> があり、それぞれ左, 右, 中央, 上, 下, ベースラインを表しています。この位置の中から 2 つまで選んで指定します。また, <code>[x=2mm, y=3mm]</code> のように座標で指定することもできます
<code>units=6.2832</code>	回転の角度の単位を, 度からラジアンに変更します。

図を入れたいところが何かしらの環境の中だったりするとエラーを吐くことがあります。そのときは、`minipage` 環境を作ってその中で `\includegraphics` をすれば大抵の場合は大丈夫です。この `minipage` 環境は横幅を引数に取り、

```
\begin{minipage}{0.4\hsize}
\centering
\includegraphics{picture.pdf}
\end{minipage}
```

などのようにして使います。この `\hsize` はページの横幅を設定しているコマンドですので、`0.4\hsize` は全ページ幅の 4 割を表しています。

## 1.1.2 図の自動配置

図の自動配置には `figure` 環境を使います。また、図にはキャプションとラベルをつけることができます。例えば以下のように、

```
\begin{figure}
\centering
\includegraphics[width=10cm]{parabola.pdf}
\caption{関数  $y = x^2$  のグラフ}
\label{fig:parabolagraph}
\end{figure}
```

とすれば、下にキャプションのついたグラフが自動的に配置されて出力されます。この図を参照するときは、`\ref{fig:parabolagraph}` において... のようにして 2 回出力すればよいです。この `\label` や `\ref`, 2 回実行する意味については相互参照やリンクを扱う第??章で詳しく扱います。

この `figure` 環境には以下のようなオプションがあります。

- t ページ上部に図を出力します。
- b ページ下部に図を出力します。
- p 単独ページに図を出力します。
- h 出来ればその位置に図を出力します。
- H 必ずその位置に図を出力します (float パッケージが必要)。

なお、このオプションはデフォルトでは [tbp] になっています。

図の番号は自動的に「図 1」, 「図 2」, ... のように付きます。これを「Fig.1」のようにしたければ, jsarticle, jsbook なら

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

のように, english オプションを付けます。それ以外のドキュメントクラスでは,

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

のようにプリアンブルに書いておきます。

\caption{図の説明} は図の説明を出力するコマンドですが, \listoffigures とすれば, その説明を使った図目次をその位置に出力することができます。説明が長いときなどは, オプションに短い説明をつけ, それを図目次に出力することもできます。図目次については, 目次を扱う章もご覧ください。

### 1.1.3 図を左右に配置する

2つの図を上下でなく左右に配置したいときもあると思います。ここではそのやり方を扱います。

関連のない2つの図を左右に配置するときは, figure 環境の中に minipage 環境を作って,

```
\begin{figure}
\centering
\begin{minipage}{0.4\columnwidth}
\centering
\includegraphics[width=\columnwidth]{figure1.pdf}
\caption{左側}
\label{fig:left}
\end{minipage}
\begin{minipage}{0.4\columnwidth}
\centering
\includegraphics[width=\columnwidth]{figure2.pdf}
\caption{右側(左と関連無し)}
\label{fig:right}
\end{minipage}
\end{figure}
```

のようにすれば, 次のようになります。



図 1.1: 左側

図 1.2: 右側 (左と関連無し)

なお, \columnwidth は本文の幅を表します。minipage 中では minipage の幅が \columnwidth になるので上のようにしています。また, minipage 自身の幅は好きに指定してください。

関連のある2つの図を左右に配置したいときは, subcaption パッケージを利用します。まずはプリアンブルに \usepackage{subcaption} と書いてください。この使い方は, 上とほぼ同じで, minipage を subfigure に置き換えて,

```

\begin{figure}
\centering
\begin{subfigure}{0.4\columnwidth}
\centering
\includegraphics[width=\columnwidth]{figure1.pdf}
\caption{左側}
\label{fig:left}
\end{subfigure}
\begin{subfigure}{0.4\columnwidth}
\centering
\includegraphics[width=\columnwidth]{figure2.pdf}
\caption{右側(左と連関無し)}
\label{fig:right}
\end{subfigure}
\caption{左右両方}
\label{fig:left-and-right}
\end{figure}

```

とします。この出力は次のようになります。



図 1.3: 左右両方

左右の図がくっついて見づらいときは、`\end{subfigure}`と`\begin{subfigure}`の間に適当に`\hspace`でも入れてください。

## 1.2 図の回り込み配置

図を文書の右とか左に配置したいときもあるかと思います。例えばこのように。そんなときは、`wrapfig` パッケージの `wrapfigure` 環境が便利です。使い方について説明します。まずはプリアンブルに

```
\usepackage{wrapfig}
```

と書いてください。例えば、上のように図を回り込ませたいときは、

```

\begin{wrapfigure}{r}{10zw}
\vspace*{-\intextsep}
\includegraphics[width=10zw]{huukei.jpg}
\end{wrapfigure}

```



のように書きます。`\begin{document}`の1つ目の引数に位置(1かr)、2つ目の引数にその幅を指定します。オプションで行数も指定できます。ただ、これは自動で計算してくれるので特に指定する必要はありません。まとめると、以下のように使うということです。

```
\begin{wrapfigure}[行数]{位置(1かr)}{幅}
```

上に示した例に`\vspace*{-\intextsep}`とありますが、これはデフォルトで入る縦方向のスペース分上に戻す、という操作をしています。これを入れるとちょうどいいって話ですね。また、参考までに、文章との間に入るスペースは`\columnsep`という長さになっています。

なお、`wrapfig` には表の回り込み配置用に `wraptable` 環境も用意されているので、必要ならば使ってみてください。また、注意ですが、この `wrapfigure` 環境は箇条書き系の環境（難しいことを言うと、ボックスを

生成するもの)と相性が悪いので、併用は避けてください。箇条書きと併用したいときは `mawarikomi` パッケージを使ってみて下さい。また、`float` パッケージを使う場合は、`float` パッケージの後に読み込んでください。

## 1.3 色

色の表し方には、RGB と CMYK の 2 通りがあります。RGB (Red, Green, Blue) はディスプレイの色に使われる表し方で、赤・緑・青の 3 色を、0~255 の 256 段階に分け、その値で色を表します。CMYK (Cyan, Magenta, Yellow, black) は印刷物に使われる表し方です。

この RGB と CMYK は互換性がなく、変換がかなり大変らしいのですが、僕はよくわからないですし、変換をしなくてもある程度似た色で印刷されるみたいなので、この話は割愛します。

色を使うときは `xcolor` パッケージを使います。まずはプリアンブル (`\documentclass` と `\begin{document}` の間) に

```
\usepackage{xcolor}
```

と書いてください。

字の色を変えることが多いと思うので、そのやり方を説明します。

グレースケールを指定するときは、

```
{\color[gray]{0.5} 文字}
\textcolor[gray]{0.5}{文字}
```

のようにします。この数値は、黒を 0、白を 1 として、0~1 で指定します。

CMYK を指定するときは、

```
{\color[cmymk]{0.75, 0.64, 0, 1} 文字}
\textcolor[cmymk]{0.75, 0.64, 0, 1}{文字}
```

のようにします。

RGB については、3 通りの表し方があります。各 RGB 値を 255 で割った値を使って、

```
{\color[rgb]{0.75, 1, 0} 文字}
\textcolor[rgb]{0.75, 1, 0}{文字}
```

のようにするのが一般的なようですが、RGB 値をそのまま使って、

```
{\color[RGB]{120, 255, 0} 文字}
\textcolor[RGB]{120, 255, 0}{文字}
```

のようにもできます。また、HTML における色の指定法を用いて、

```
{\color[HTML]{10A3FF} 文字}
\textcolor[HTML]{10A3FF}{文字}
```

のようにもできます。

いちいち色を数値で指定するのは面倒ですので、いくつかの色はもともと定義されており、その名前を書くだけで使えます。また、`xcolor` パッケージに `dvipsnames` オプションを付けると、もっと多くの色名を使うことができます。

しかし、これだけでは欲しい色が手に入らない、という場合もあると思います。そのときは、次のようにして色を定義します。

```
\definecolor[gray]{x}
\definecolor[rgb]{r, g, b}
\definecolor[cmymk]{c, m, y, k}
```

RGB については、文字色の変更のときと同様、オプションを RGB や HTML にして、それに従った記法で定義することもできます。

元々ある色名や定義した色名を使って、次のように色を指定することもできます。

```
{\color{色名} 文字}  
\textcolor{色名}{文字}
```

なお、`\color` コマンドは、適用範囲を `{}` で囲まないとそれ以降全ての文字の色が変わってしまうので注意が必要です。

ページの色を変えるには、

```
\pagecolor{色名}
```

とします。それ以降のページは全てその色になるので、白に戻すには `\pagecolor{white}` とします。文字の背景に色を付けたいときは、

```
\colorbox{色名}{文字}
```

とします。なお、色名はオプションと数値でも指定できます。例えばこんな風にこれに枠を付けたいときは、

```
\fcolorbox{枠色}{背景色}{文字}
```

とします。 わくわく