

修士論文 2019 年度（令和元年度）

手書きベース Wiki システムの研究

慶應義塾大学大学院 政策・メディア研究科

早川 匠

2019 年 1 月

修士論文 2019 年度（平成元年度）

手書きベース Wiki システムの研究

論文要旨

手書きメモやイラスト等のグラフィカルなデータに自在にハイパーリンクを埋め込み、それらを Wiki として利用できるシステム「手書きベース Wiki」を提案する。手書きメモやイラストは広く浸透した情報の記録・表現手法であるものの、紙を前提としたフォーマットであるために参照や管理や再利用が難しいという問題が存在する。計算機上で手書きメモの作成や管理を行うツールは広く利用されているものの、これらは紙のメモやイラストの利用形態を再現したに過ぎず、この問題を本質的に解決していない。「手書きベース Wiki」とはハイパーテキスト・ハイパーリンクや Wiki 等の技術の組み合わせによって、簡単に手書きでメモやイラストを描きながら、自在にハイパーリンクを埋め込んだり、ハイパーリンクによって関連する他のメモやイラストを簡単に参照できるシステムである。これによって既存の手書きメモ・イラストの問題点が解決されるだけでなく、新しい活用法が提案できる。本論文では「手書きベース Wiki」の設計や実装、その応用例について述べ、研究の発展性について考察する。

キーワード

手書きメモ、イラスト、Wiki、ハイパーテキスト、ユーザーインターフェース

慶應義塾大学大学院 政策・メディア研究科

早川 匠

Abstract Of Master's Thesis Academic Year 2019

A study on drawing-based wiki systems

Summary

Eigo ga dekinai node Roma-ji de soreppoi hunniki wo daseruto iina.

Murippoi desu ne.

Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here.

Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here. Write down your abstract here.

Write down your abstract here. Write down your abstract here.

Keywords

Handwritten notes, illustration, Wiki, Hypertext, User Interface

Graduate School of Media and Governance
Keio University

Takumi Hayakawa

目次

第1章	序論	1
1.1	研究の動機	2
1.2	研究の目的	2
1.3	本論文の構成	3
第2章	本テンプレートの使い方	5
2.1	テンプレートの構成	5
2.2	コンパイル	5
2.3	設定	6
2.3.1	論文全体の言語の設定	6
2.3.2	余白の設定	6
2.3.3	論文情報の設定	7
2.4	出力	8
2.4.1	外部ファイルの読み込み (<code>include</code>)	8
2.4.2	表紙の出力	9
2.4.3	アブストラクトの出力	9
2.4.4	目次類の出力	10
2.4.5	本文の出力	10
2.4.6	謝辞の出力	11
2.4.7	参考文献の出力	11
2.4.8	付録の出力	13
第3章	L^AT_EX の書き方	15
3.1	主なコマンド	15
3.1.1	章と節	15
3.1.2	図	16
3.1.3	表	18
3.1.4	脚注	20
3.2	その他のコマンド	20
第4章	結論	21
4.1	まとめ	21
4.2	大事なこと	21

謝辞	22
参考文献	24
付 録 A 付録の例	27
A.1 ほげ	27
A.1.1 ふー	27
A.2 ほげほげ	27
A.2.1 ふーふー	27

図 目 次

3.1 図の例	16
3.2 図を並べる例 1	18
3.3 図を並べる例 2、枠なし	18

表 目 次

2.1	構成ファイル	5
3.1	章と節のコマンド	16
3.2	表の例	19
3.3	セルを結合した例	20

第1章 序論

本章では本研究の動機と目的、および本論文の構成について述べる。

1.1 研究の動機

手書きでメモを取ったり絵や図を描いたりすることは、情報の記録し、表現する手段として一般的であるが、その様式は紙や鉛筆等の筆記具が発明された頃からほとんど変わっておらず、一枚の紙の上で表現する事を前提としているため参照や管理、再利用が難しいという問題点が存在する。

また計算機が普及した現在では、手書きメモ・イラストを紙の上ではなくデータとして作成・記録し、それらを閲覧・管理できるソフトウェアも広く利用されているが、そのデータは JPEG や PNG のような一枚の絵をピクセルの集合で単純に置き換えた形式で作成・記録されることが一般的で、紙に描かれたものと比較して本質的に変化していない。そのため、データのタイムスタンプを元に時系列で管理したり、あるいは事前に取り決めた場所 (ディレクトリ) に保存する等の工夫が必要とされ、紙の上で手書きのメモを取っていた時と同質の不便さを引き継いでいる。

一方でメモやイラストと同じく紙の上で記録されていたテキストは、計算機の登場により以下のような機能を獲得した。

- コピー・ペースト/Undo・Redo 等の編集支援機能
- ハイパーリンクと Web による他のテキスト (ドキュメント) への参照
- 複数人による共同編集を可能にする Wiki 等のコラボレーションツール

これにより (1) 簡単に編集できない、(2) 参照や管理が面倒という問題が解決された。かつては手書きメモ・イラストと同様の問題を抱えていたテキストは、計算機による新しい活用法が発明された事で広く普及するに至った。そのため手書きメモ・イラストの活用も、計算機を活用する事で同様の進化が可能であると考えられる。

1.2 研究の目的

本研究では、手書きメモ・イラストを表現する既存のフォーマットが抱える問題を解決し、またハイパーテキストや Wiki の手法を取り入れることで、従来のシステムでは実現できなかった手書きデータの参照・管理・再利用環境を実現するシステム「手書きベース Wiki」の構築を目的とする。

1.3 本論文の構成

本論文は以下の 8 章で構成される。

第 2 章では、本研究の背景をより詳細に分析し、既存システムの問題点を整理する。

第 3 章では、本論文で提案するシステムの基本構成と使い方について述べる。

第 4 章では、本論文で提案するシステムの詳細な実装について述べる。

第 5 章では、本論文で提案するシステムによって実現可能な応用例について述べる。

第 6 章では、関連する研究を紹介し、それらの特徴や本研究との関連を述べる。

第 7 章では、筆者による運用経験やユーザーからのフィードバックをまとめ、本論文で提案するシステムの有効性と問題点について述べる。

最後に、第 8 章で本論文のまとめと結論を述べる。

第2章 本テンプレートの使い方

本章では、本テンプレートの具体的な使用方法を解説する。基本的には、`main.tex` を上から順に修正していけばよいだけ。

2.1 テンプレートの構成

このテンプレートは、表 2.1 のファイルで構成されている。

表 2.1: 構成ファイル

ファイル名	用途
<code>main.tex</code>	メインのファイル。これを編集していく
<code>thesis.sty</code>	論文のスタイルを定義したファイル。基本的には手は加えない
<code>*.tex</code>	<code>main.tex</code> に <code>include</code> されるファイル群
<code>*.eps</code>	画像ファイル
<code>main.bib</code>	参考文献用の BibTeX ファイル
<code>Makefile</code>	Makefile。次節以降で説明
<code>.gitignore</code>	Git 用設定ファイル

2.2 コンパイル

このテンプレートの L^AT_EX ファイルをコンパイルして PDF ファイルを生成するには、ターミナルを開いて以下のようにする。

コマンド実行例

```
% make
```

こうすることで、`platex` コマンド、`pbibtex` コマンド、`platex` コマンド 2 回、`dvipdfmx` コマンドが全て実行され、`main.pdf` が生成される。

コンパイルによって生成されたファイルを全て消すには、以下のようにする。

コマンド実行例

```
% make clean
```

2.3 設定

以下、`main.tex` に対して行うべき設定を、このファイルの中に書いてある順に沿って説明する。

2.3.1 論文全体の言語の設定

`main.tex`

```
\japanesetrue % 論文全体を日本語で書く（英語で書くならコメントアウト）
```

ここでは論文全体の言語を設定する。日本語に設定すれば、『章』『目次』『謝辞』などが日本語で出力されて、行頭のインデントなども日本語の仕様になる。英語にした場合は、これらはそれぞれ『Chapter』『Table of Contents』『Acknowledgment』な体裁になる。インデントも行間も、英語用の設定が適用される。

`\japanesetrue` をコメントアウトしなければ日本語に、コメントアウトすれば英語に設定される。

2.3.2 余白の設定

`main.tex`

```
\bindermode % バインダ用余白設定
```

このテンプレートの出力は A4 用紙。ここではこれの四辺の余白を設定する。

最終的にバインダーで綴じて提出する場合、余白を左右対称にしてしまうと、見かけ上のバランスがとても悪くなる。これを解消するため、あらかじめ左側の余白を大きく取っておく。

`\bindermode` をコメントアウトしなければ左綴じ用の余白に、コメントアウトすれば左右対称の余白に設定される。

両面印刷の場合、偶数ページと奇数ページで余白を広くとるべき側が違うので、`documentclass` でこれを設定する。

main.tex

```
% 両面印刷の場合。余白を綴じ側に作って右起こし。
\documentclass[a4j,twoside,openright,11pt]{jreport}

% 片面印刷の場合。
%\documentclass[a4j,11pt]{jreport}
```

両面印刷の場合は `twoside` を使用する。`openright` を使うと章のはじまりが必ず右側のページに来るようになる。

2.3.3 論文情報の設定

main.tex

```
% 日本語情報（必要なら）
\jclass {修士論文} % 論文種別
\jtitle {修士論文用 \LaTeX\ テンプレート} % タイトル。改行する場合は
\\を入れる
\juniv {慶應義塾大学大学院} % 大学名
\jfaculty {政策・メディア研究科} % 学部、学科
\jauthor {ほげ山 ふう助} % 著者
\jhyyear {24} % 平成〇年度
\jsyear {2012} % 西暦〇年度
\jkeyword {\LaTeX、テンプレート、修士論文} % 論文のキーワード
\jproject{インタラクシオンデザインプロジェクト} %プロジェクト名
\jdate{2013 年 1 月}

% 英語情報（必要なら）
\eclass {Master's Thesis} % 論文種別
\etitle {A \LaTeX Template for Master Thesis} % タイトル。改行す
る場合は\\を入れる
\euniv {Keio University} % 大学名
\efaculty {Graduate School of Media and Governance} % 学部、学科
\eauthor {Fusuke HogeYama} % 著者
\eyear {2012} % 西暦〇年度
\ekeyword {\LaTeX, Template, Master Thesis} % 論文のキーワード
\eproject{Interaction Design Project} %プロジェクト名
\edate{January 2013}
```

ここでは論文のタイトルや著者の氏名などのメタデータを記述する。ここで書いたデータは、表紙とアブストラクトのページに使われる。必ずしも日本語と英語の両方を設定しなけ

ればいけないわけではなくて、自分が必要とする方だけ記述すればよい。

タイトルが長過ぎる場合は、表紙やアブストラクトのページでは自動で折り返して出力される。もし改行位置を自分で指定したい場合は、その場所に `\\` を入力する。

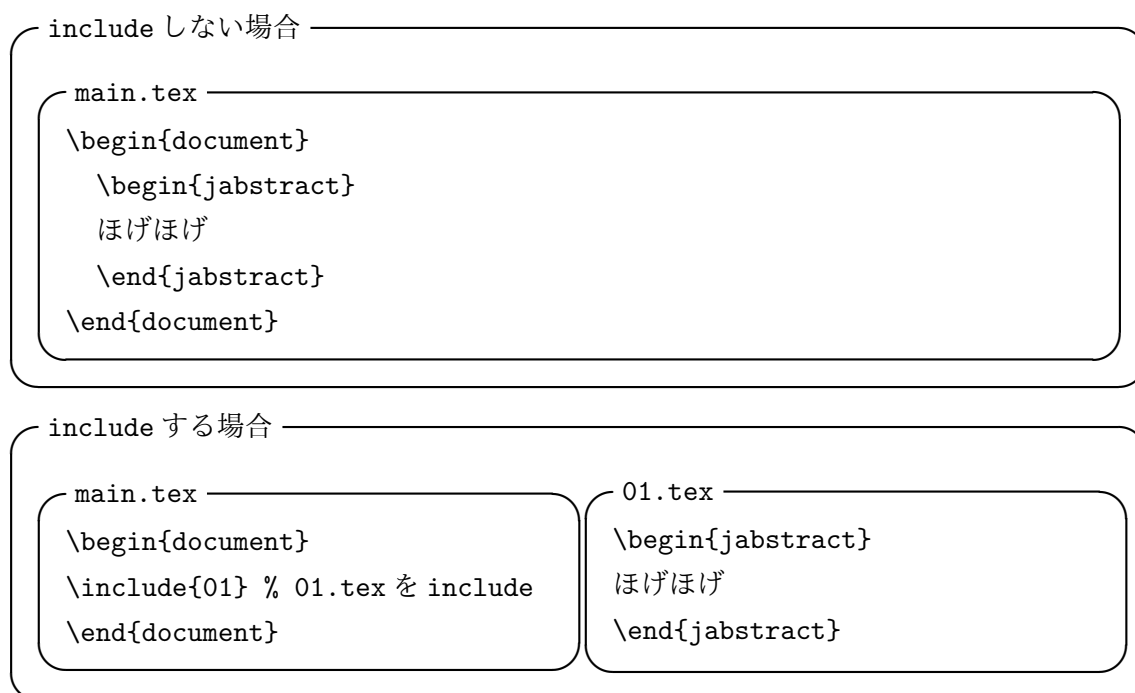
2.4 出力

`\begin{document}` から `\end{document}` に記述した部分が、実際に DVI（最終的には PDF）ファイルとして出力される。

2.4.1 外部ファイルの読み込み（include）

出力部分の具体的な説明の前に、外部ファイルを読み込む方法を説明する。

`\begin{document}` から `\end{document}` の間では、`\include` コマンドを使うことで、別の `*.tex` ファイルを読み込ませられる。



`\include` しない場合とする場合を比較するとこのとおり。どちらも出力結果は一緒。`\include` する場合は、読み込ませたい箇所に、読み込ませたい `*.tex` ファイルの名前を、括弧子を除いて `\include` コマンドで書けばよい。

`\include` コマンドを用いるか用いないかは、たぶん文書量や個人の好みに依る。例えば章ごとに別のファイルにしておけば、修正箇所を探すときの手間が多少は省けるかもしれない。Git で人と共有しつつ校正を頼むときにもファイルが分かれていたほうがコンフリクトを起こしにくい。

2.4.2 表紙の出力

```
main.tex
\ifjapanese
  \jmaketitle    % 表紙（日本語）
\else
  \emaketitle    % 表紙（英語）
\fi
```

最初に、表紙を出力する。

`\jmaketitle` が実行されると日本語の表紙が、`\emaketitle` が実行されると英語の表紙がそれぞれ出力される。日本語の表紙には、第 2.3.3 節で設定したうちの日本語の情報が、英語の表紙には同節で設定したうち英語の情報が、それぞれ参照されて、表記される。

デフォルトでは第 2.3.1 説で設定した言語の表紙のみが出力されるようになっている。

2.4.3 アブストラクトの出力

```
main.tex
\include{00_abstract} % アブストラクト。要独自コマンド、include 先参照のこと
```

表紙の次は、アブストラクト。

アブストラクトを出力するには、出力したい位置に、指定のコマンドを用いて文章を書き下せばよい。`main.tex` に直接書いてもよいし、先述した `\include` コマンドを利用して `include` してもよい。

`\begin{jabstract}` から `\end{jabstract}` の間に書いた文章が日本語のアブストラクトとして、`\begin{eabstract}` から `\end{eabstract}` の間に書いた文章が英語のアブストラクトとして、それぞれ独立したページに出力される。

アブストラクトのページには、論文のタイトルやキーワードなどが、第 2.3.3 節で設定した情報をもとにして自動で表記される。

日本語か英語のどちらか一方のみでよい場合は、不要な言語の方のコマンドを削除すればよい。これは、`\begin` と `\end` というコマンド自身も含めて削除する、ということで、`\begin` と `\end` の間を空っぽにするという意味ではないので注意。

2.4.4 目次類の出力

```
main.tex
\tableofcontents % 目次
\listoffigures % 表目次
\listoftables % 図目次
```

アブストラクトの次に、目次。文書の目次、図の目次、表の目次の三種類。

目次類を出力するには、出力したい位置に指定のコマンドを書けばよい。

これらのコマンドは、コンパイル時点での一時ファイル¹の情報を、目次として体裁を整えて出力するもの。一時ファイルは、`\begin{document}` から `\end{document}` の間の章や節、図や表をコンパイルするときに、ついでに情報を取得しておいて生成される。

つまり気をつけなければいけないのは、コンパイルを一回しただけでは、一時ファイルが最新の状態に更新されるだけで、肝心の目次は正しい情報では出力されないということ。目次類を正しい情報で出力するには、最低二回のコンパイルが必要。一回目のコンパイルで一時ファイルが最新の情報に更新されて、二回目のコンパイルで初めて、その最新の一時ファイルの情報をもとに目次が出力される。

だから、文書に何らかの修正をして保存したあとは、最低でも二回、連続してコンパイルしないとイケないことに注意する。

図や表を一つも使用していない場合は、目次名のみが書かれた空白のページが出力される。もしこれが不要な場合は、該当するコマンドをコメントアウトすればよい。

2.4.5 本文の出力

```
main.tex
\include{01} % 本文 1
\include{02} % 本文 2
\include{03} % 本文 3
\include{04} % 本文 4
```

目次に続いて、論文のメイン、本文を記述する。アブストラクトと同様で、`main.tex` に直接書くか、`\include` コマンドを利用して別に用意したファイルを `include` する。

本文の書き方は、第3章で詳しく説明する。

¹*.toc、*.lof、*.lot

2.4.6 謝辞の出力

```
main.tex —————  
\include{90_acknowledgment} % 謝辞。要独自コマンド、include 先参照のこと
```

本文のあとには、謝辞を出力する。`begin{acknowledgment}` から `end{acknowledgment}` の間に書いた文章が、謝辞として独立したページに出力される。アブストラクトや本文と同じで、`main.tex` に直接書いてもよいし、`\include` コマンドを利用して `include` してもよい。

2.4.7 参考文献の出力

```
main.tex —————  
\include{91_bibliography} % 参考文献。要独自コマンド、include 先参照のこと
```

謝辞に続いて、参考文献を出力する。

参考文献リストは、`\begin{bib}` から `\end{bib}` の間に、`\bibitem` コマンドを使って書く。

BibTeX を使う場合は、以下のようにする。

```
91_bibliography.tex —————  
\begin{bib}[100]  
\bibliography{main}  
\end{bib}
```

こうすると、`main.bib` から使用した参考文献のみを抽出して出力してくれる。`main.bib` の中身は以下のようになっていて、気の利いた論文検索サイトであれば BibTeX をコピペでできるようになっているので簡単に作れるはず。

91_bibliography.tex

```
@article{hoge09,
  author = "ほげ山太郎 and ほげ山次郎",
  yomi    = "ほげやまたろう",
  title   = "ほげほげ理論の HCI 分野への応用",
  journal = "ほげほげ学会論文誌",
  volume  = "31",
  number  = "3",
  pages   = "194-201",
  year    = "2009",
}

@inproceedings{hoge08,
  author      = "Taro HogeYama and Jiro HogeYama",
  title       = "The Theory of Hoge",
  booktitle   = "The Proceedings of The Hoge Society",
  year        = "2008"
}
```

以下は、BibTeX を使わないで手で書く例。

91_bibliography.tex

```
@article{hoge09,
  author = "ほげ山太郎 and ほげ山次郎",
  yomi    = "ほげやまたろう",
  title   = "ほげほげ理論の HCI 分野への応用",
  journal = "ほげほげ学会論文誌",
  volume  = "31",
  number  = "3",
  pages   = "194-201",
  year    = "2009",
}

@inproceedings{hoge08,
  author      = "Taro HogeYama and Jiro HogeYama",
  title       = "The Theory of Hoge",
  booktitle   = "The Proceedings of The Hoge Society",
  year        = "2008"
}
```

英語の文献の場合、慣例的に書誌名をイタリック体にすることが多いらしい。

```

91.bibliography.tex
\begin{bib}[100]
\begin{thebibliography}{#1}
% \bibitem{参照用名称}
%   著者名 :
%   \newblock 文献名,
%   \newblock 書誌情報, 出版年.

\bibitem{hoge09}
  ほげ山太郎, ほげ山次郎 :
  \newblock ほげほげ理論の HCI 分野への応用,
  \newblock ほげほげ学会論文誌, Vol.31, No.3, pp.194-201, 2009.

\bibitem{hoge08}
  Taro HogeYama, Jiro HogeYama:
  \newblock The Theory of Hoge,
  \newblock {\it The Proceedings of The Hoge Society}, 2008.
\end{thebibliography}
\end{bib}

```

`\bibitem` コマンド中、参照用名称は、本文から参考文献を参照するときに使うので、忘れずに書いておく。参考文献を本文中に参照するときには、`\cite{参照用名称}` のように書けばよい。例えば、この文の末尾には `\cite{hoge09}` と書いてあるので、自動で対応する番号が振られる [2][1]。

参考文献リストの番号付けと、本文で参照したときの番号の挿入は、全部が自動で行われる。ただしこれも、第 2.4.4 節で説明した目次の出力と同じで、一時ファイルを生成してからの挿入なので、正しく出力するには最低でも二回のコンパイルが必要。BibTeX を使用する場合は、`platex` コマンドのあと `pbibtex` コマンドを実行し、さらに 2 回 `platex` コマンドを実行するといいらしい。

2.4.8 付録の出力

```

main.tex
\appendix
\include{92_appendix} % 付録

```

必要であれば、論文の最後には付録を出力する。

`\appendix` コマンド以降に書いたものは、すべて付録として扱われる。付録部分の書き方は通常の本文とまったく同じで、`\appendix` コマンド以降に書くだけで勝手に付録用の

体裁で出力される。

第3章 L^AT_EX の書き方

この章では、よく使う L^AT_EX のコマンドを説明する。足りない部分はぐぐればだいたいわかると思う。最初に書いておくと、数式を書く方法は、ぼく自身使わなかったので書いていない。ぼくのいた研究室でゴリゴリ数式をたくさん書く必要のあるひとは、研究の種類からするとあまり居ない気がする。

3.1 主なコマンド

3.1.1 章と節

文書構造を明確にする大事なもの。目次はこれらのコマンドをもとに作られる。例えば、この第3章の冒頭部分はこのようなソースで書かれている。

03.tex

```
\chapter{\LaTeX の書き方}
\label{chap:latex}
```

この章では、よく使う \LaTeX のコマンドを説明する。(略)

```
\section{主なコマンド}
```

```
\subsection{章と節}
```

文書構造を明確にする大事なもの。目次はこれらのコマンドをもとに作られる。例えば、この第\ref{chap:latex}章の冒頭部分はこのようなソースで書かれている。

章は \chapter{見出し}、節は \section{見出し}、小節は \subsection{見出し}、小々節は \subsubsection{見出し} を使う。表 3.1 に一覧する。

(1) 小々節見出しサンプルその 1

小々節は上のように \subsubsection{タイトル} で書けるけれど、あまり文書の階層構造が深いことは望ましくないなので、多用しなければならないような文書構造を見直したほうがよいと思う。

表 3.1: 章と節のコマンド

コマンド	用途
<code>\chapter{見出し}</code>	章
<code>\section{見出し}</code>	節
<code>\subsection{見出し}</code>	小節
<code>\subsubsection{見出し}</code>	小々節

(2) 小々節見出しサンプルその 2

小々節は、章や節、小節のように N.N.N といった番号ではなくて、括弧付きの番号で出力される。かつ、目次には出力されない。

3.1.2 図

図は次のように出力される（図 3.1）。

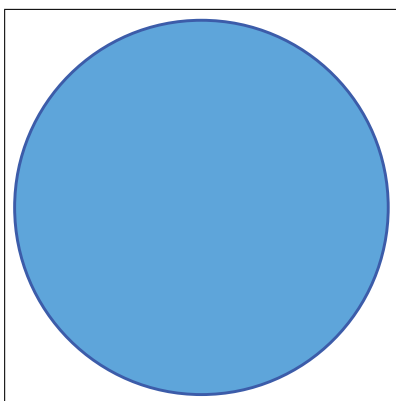


図 3.1: 図の例

ソースでは次のように記述している。

03.tex

図は次のように出力される（図\ref{fig:sample1}）。

```
\begin{figure}[htbp]
  \begin{center}
    \fbox{\includegraphics[width=40mm]{image.eps}}
  \end{center}
  \caption{図の例}
  \label{fig:sample1}
\end{figure}
```

`\begin{figure}[htbp]` の `htbp` は、表示位置の優先順位の設定。基本的に \LaTeX では、図の挿入位置は強制的には指定できない。いくつか候補を指定しておく、候補のなかの優先度の高い順に、図を入れられるスペースがあるかどうかを調べて、入れられればそこに、入れられなければ次の候補のスペースを調べる、という処理が行われる。`h` はこのコマンドを書いたその場所に、`t` はページの一番上に、`b` はページの一番下に、`p` は画像だけ別ページに、それぞれ配置する。基本的には `htbp` のように全部書いておけば問題ない。

`\includegraphics` コマンドで、図のサイズと挿入するファイルを指定する。上の例ではサイズは `width=50mm` として幅を指定したけれど、ここは他にも `height=30mm` として高さを指定してもよいし、`scale=0.5` として拡大率を指定してもよい。画像は最近の \LaTeX 環境であれば `*.eps` 以外でも使える。ただし、`bb` (Bounding Box) として画像の大きさを指定する必要があることも多い。以下は JPEG 画像を使用する例。

03.tex

```
\begin{figure}[htbp]
  \begin{center}
    \fbox{\includegraphics[width=40mm,bb=0 0 640 480]{image.jpg}}
  \end{center}
  \caption{図の例}
  \label{fig:sample1}
\end{figure}
```

`bb` の指定は、上記のように `*.tex` ファイルの中で指定してもいいが、`*.bb` ファイルを作っておく方法もある。ターミナルで `ebb` コマンドを使用すると `*.bb` ファイルを簡単に作れる。

ebb コマンドの例

```
% ebb image.jpg
```

`\includegraphics` を `\fbox` に入れると、画像に枠を付けられる。

`\caption` コマンドで図の見出しを指定できる。図の見出しは、図の下に表記するので注意。ここで指定した見出しが、図の目次に表示される。

`\label` コマンドでは図の参照用ラベルを設定できる。本文中、`\ref` コマンドで参照用ラベルを指定すると、対応した図の番号が自動的に挿入される。これも目次や参考文献と同様、最低二回のコンパイルが必要なので注意。

図を二つ横に並べたい場合は、次のように書く（図 3.2、図 3.3）。

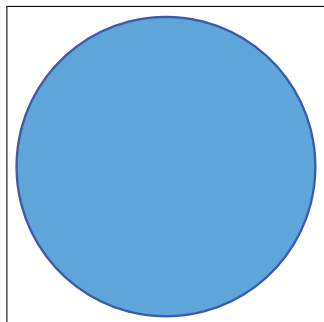


図 3.2: 図を並べる例 1

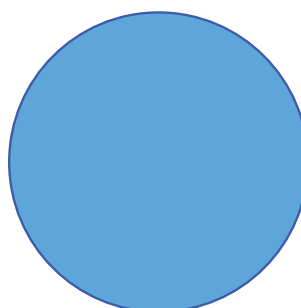


図 3.3: 図を並べる例 2、枠なし

03.tex

図を二つ横に並べたい場合は、次のように書く（`\ref{fig:sample2}`、`\ref{fig:sample3}`）。

```
\begin{figure}[htbp]
  \begin{minipage}{0.5\hsize}
    \begin{center}
      \fbox{\includegraphics[width=40mm]{image.eps}}
    \end{center}
    \caption{図を並べる例 1}
    \label{fig:sample2}
  \end{minipage}
  \begin{minipage}{0.5\hsize}
    \begin{center}
      \fbox{\includegraphics[width=40mm]{image.eps}}
    \end{center}
    \caption{図を並べる例 2}
    \label{fig:sample3}
  \end{minipage}
\end{figure}
```

3.1.3 表

表は次のように出力される（表 3.2）。

表 3.2: 表の例

種類	味	評価
ドラ焼き	甘い	好き
メロンパン	カリもふ	好き
クリームパン	神	すごく好き

ソースでは次のようになっている。

03.tex

表は次のように出力される（表`\ref{tb:sample1}`）。

```
\begin{table}[htbp]
  \caption{表の例}
  \label{tb:sample1}
  \begin{center}
    \begin{tabular}{l|c|r}
      \hline
      種類 & 味& 評価\\\hline\hline
      ドラ焼き& 甘い& 好き\\\hline
      メロンパン& カリもふ& 好き\\\hline
      クリームパン& 神& すごく好き\\\hline
    \end{tabular}
  \end{center}
\end{table}
```

`htbp` や `\caption` と `\label` は図と同様。ただし表のタイトルは表の上を書く。

`\begin{tabular}{l|c|r}` で横方向のセルを指定する。`c` は中央揃え、`l` は左揃え、`r` は右揃えのセルを作る。`|` は垂直方向の罫線を表す。`c` か `l` か `r` を必要なセルの数だけ並べて、セルの間に罫線が必要なら `|` を入れればよい。

セルの中の文字は、`&` で区切って並べる。行と行は `\\` で区切る。水平方向の罫線が必要なら、`\hline` を書く。

水平方向や垂直方向のセルの結合もできる。例を示すので、くわしくはぐぐろう。説明がめんどう。`\multirow`、`\multicolumn`、`\cline` を使うとできる。

表 3.3: セルを結合した例

ほげ	ふー	ばー
ほげほげ	ふーふー	
	ふーふーふー	ばーばーばー

03.tex

```
\begin{table}[htbp]
  \caption{セルを結合した例}
  \label{tb:sample2}
  \begin{center}
    \begin{tabular}{c|c|c}
      \hline
      ほげ&ふー&ばー\\\hline\hline
      \multirow{2}{*}{ほげほげ}&\multicolumn{2}{c}{ふーふー} \\
      &ふーふーふー&ばーばーばー\\\hline
    \end{tabular}
  \end{center}
\end{table}
```

3.1.4 脚注

脚注は `\footnote` コマンドを使う。例えばこんな感じ¹。

03.tex

例えばこんな感じ `\footnote{ページの下に小さく説明を出せる}`。

3.2 その他のコマンド

ぐるぐる²。

特殊なことは何もしていないテンプレートなので、ぐるぐるって出たことはだいたいそのまま何でも使える。

あるいは、このファイル自体も `LATEX` で書かれているわけだから、これの `*.tex` を見るのもよいかもしれない。

¹ページの下に小さく説明を出せる

²<http://www.google.co.jp/>

第4章 結論

この章では、結論らしいことをかく。

4.1 まとめ

L^AT_EX の環境さえあればスタンダードな体裁の論文がたぶんだれでも作れる程度のテンプレートにはなっているはず。がんばって卒業しよう。

4.2 大事なこと

箇条書きで列挙する。

- ぐぐる。これは単なる L^AT_EX だし、L^AT_EX はもう枯れた技術だから、調べれば文献はいくらでもある。
- 先生を頼る。
- 単位をきちんととる。
- 卒業する。

謝辞

このテンプレートを改造するにあたって、@kurokobo とインターネット上のいくつかの修士論文などを参考にしました。感謝いたします。

参考文献

- [1] Taro HogeYama and Jiro HogeYama. The theory of hoge. In *The Proceedings of The Hoge Society*, 2008.
- [2] ほげ山太郎, ほげ山次郎. ほげほげ理論の hci 分野への応用. ほげほげ学会論文誌, Vol. 31, No. 3, pp. 194–201, 2009.

付 録 A 付録の例

付録を無理矢理出力させるため、てきとうなことを書く。

A.1 ほげ

コマンドは本文と一緒に。

A.1.1 ふー

本文と一緒に。

A.2 ほげほげ

本文と一緒に。

A.2.1 ふーふー

本文と一緒に。