

LaTeX 文書作成への招待

水谷正大

2016 年 12 月 4 日

はじめに

原稿用紙に文章を書くだけで済んでしまう人は作家などごく限られた人達だけです。特にコンピュータ利用が一般的になっている今日では、書いた文章が読む側にとって快適で読みやすいように丁寧に配慮された文書として完成度を高めることが求められています。

いうまでもなくレポートや本は内容が大切です。私たちも小説家も読み甲斐のある内容とするために本文を何度も推敲するのはそのためです。『文は人なり』という言葉にあるように書かれている一文一文は著者の何かを濃厚に反映しています。しかし、いくら中身が立派であってもそれだけでは良い読み物にはなりません。

手書きから印刷物となって以来、長い年月を経て洗練されてきた印刷物の品質やレイアウト上の配慮が追求され続けてきました。文字だけでなく数式、化学式や楽譜などの記号や図表を混在させた文書としての美しさと可読性と品質への要求はタブレットなど電子画面と文書閲覧の方法やその配布方式が多様化してきた現在においてもますます高まっています。

L^AT_EX は DEC のコンピュータ科学者であった Leslie Lamport が Stanford 大学の Donald Knuth が開発した T_EX を誰でも使えるように使いやすいようにマクロパッケージ化した電子組版システムです^{*1}。

L^AT_EX システムが文書作成において優れている理由として

- 印刷文書作成支援システムとして完成度が高く、組版作業が自動化できる。
- 目次生成、章・節や文献番号および図表番号の相互参照や索引など容易に行うことができ、リアルタイムで正確な文書作成を可能にする。
- 文書ファイルが編集・文書処理が可能なテキストファイルでありながら、出力結果とファイルが 1 対 1 に対応している
- 多くの文書様式のためのスタイルファイルが用意されており、同じ文書内容であってもスタイルファイルを変更するだけで文書レイアウトを変えることができる。

などがあげられます。

美しく読みやすい文書、とくに長大な文書がどのようなものであり、どのように作成すればよいかに対する回答として L^AT_EX を使うことを紹介します。L^AT_EX がワードプロセッサを利用する場合と何が違っており、どんな優位性あるいは課題があるのかを理解することは文書作成に対する大きな指針となるでしょう。

^{*1} Donald Knuth は 1974 年に、Leslie Lamport は 2013 年に、いずれもコンピュータ科学に対する大きな貢献に対して Turing 賞が授与されました。

目次

第 1 章	L^AT_EX をはじめよう	1
1.1	L ^A T _E X で組版する	1
1.2	文章の基本を確認する	4
1.3	特別な文字の取り扱い	6
第 I 部	文書全体についての考え方	8
第 2 章	文書を構成する 文章 ≠ 文書	9
2.1	表題の出力	9
2.2	見出しを付ける = 構造化	10
2.3	目次を作成する	11
2.4	文書構造と文書クラス	11
2.5	文書作成の進め方	13
2.6	長大な文書作成ではファイルを分割する	16
第 3 章	文書のレイアウト	18
3.1	書式と環境	18
3.2	文書の概要	18
3.3	文の引用	18
3.4	文を寄せる	19
3.5	擬似タイプ入力	21
3.6	簡条書	21
3.7	ネストされた環境	23
3.8	脚注と傍注	25
3.9	L ^A T _E X で使われる文字	26
3.10	シンボル・特殊記号の表現	27
第 II 部	それぞれのテクニック	30
第 4 章	正確な文書と相互参照	31
4.1	参考文献リストの作成	31
4.2	Web ページ情報の参照	32

4.3	参考文献の品質	37
4.4	相互参照を使った正確な文書	37
4.5	正確な文書作成はなぜ必要なのか	40
第 5 章	作表と画像の取り込み	42
5.1	簡単な作表	42
5.2	画像ファイルの取り込み	46
第 6 章	数式を書く	49
6.1	2 つの数式モード	49
6.2	簡単な数式	49
6.3	数式を組み立てる	49
第 7 章	索引の作成	50
第 III 部	さらなる応用	53
第 8 章	パッケージの利用	54
8.1	段組文書	54
8.2	パッケージの入手とインストール	55
8.3	PDF ファイルのハイパーリンク	56
8.4	TEXMFHOME の利用	56
8.5	描画パッケージ TikZ	59
第 9 章	縦組文書	60
9.1	縦書き文書クラス	60
9.2	ルビをふる	60
9.3	脚注	61
第 10 章	スライドおよびポスターの作成	63
10.1	Beamer でスライド	63
10.2	Beamer でポスター	66
第 IV 部	さらに進む	68
第 11 章	文書作成再考	69
11.1	文書スタイルと組版エンジン	69
11.2	多言語文書の作成	69
付録 A	L^AT_EX 情報を調べる	70
付録 B	L^AT_EX のクラウドサービス	71

B.1	ShareLaTeX	71
B.2	CloudLaTeX	73
B.3	Overleaf	74
付録 C	L ^A T _E X のインストール	75
付録 D	L ^A T _E X 処理の流れ	76
D.1	L ^A T _E X システムの使い方	76
D.2	L ^A T _E X システムの作業の流れ	76
D.3	組版出力までの手順	78
付録 E	Web ページで L ^A T _E X 記法を使う	85
参考文献		86

第 1 章

L^AT_EX をはじめよう

1.1 L^AT_EX で組版する

L^AT_EX システムでまず**組版** (type set) してみましょう。組版とはページ毎に文字や図表などを読みやすく配置する作業です。組版ということばにまだ慣れていないかもしれませんが、L^AT_EX では組版が自動化され、作成する文書全体で首尾一貫したスタイルが維持されます。どのようなスタイルで文書を作成すれば読みやすくなるかについては、印刷技術の発達とともに膨大なノウハウが蓄積されており、さらに出版社などで独自のデザインを加味しています。

L^AT_EX ではこの組版スタイルを**文書クラス**といい、十分に納得できるクラスが用意されてます。本書では、主に [1] の著者である奥村晴彦先生が開発した広く使われている日本語論文クラス jsarticle を使った文書作成を紹介します。

L^AT_EX システムがパソコンにインストールされていなくても、付録 B で紹介したクラウドサービスに登録すればどこからでも L^AT_EX で組版できます。是非この機会に無料登録してみてください。幸運にも L^AT_EX がインストールされて環境設定がなされたパソコンがあれば以下のようにしてみましょう。L^AT_EX システムのインストールや設定については付録 C で紹介しています（設定も自動化されているのでほぼ何もする必要はありません）。

図 1.1 は L^AT_EX 統合ソフトウェア TeXWorks、図 1.2 は Mac 用統合ソフトウェア TeXShop を使って組版する様子を示しています。ここでは組版エンジンとして“pLaTeX”が選択されているとします（組版エンジンについては第 11 章や付録 C および付録 D.1 で説明しています）。組版ボタン押すと組版処理が始まり、エラーを生じなければ組版された印刷イメージが別ウインドウに現れます。一連の組版処理を実行して印刷イメージを得るまでの作業を**ビルド** (build) ということもあります。

この例ではウインドウに入力されたテキスト内容をファイル名 `my_first_document.tex` で保存しています（お勧めの文字符号化は UTF-8 で、L^AT_EX システムは UTF-8 のソースファイルを組版するように設定されているとします）。これを L^AT_EX ファイルまたは**ソースファイル** (source file) といい、文書内容と組版のあり方を指示する原稿に相当します。L^AT_EX ソースファイルは**拡張子** ‘.tex’ が必要です。ソースファイル内の記号文字 ‘\’ は**バックスラッシュ** (backslash) です。バックスラッシュは日本語 Windows ではしばしば記号 ‘\’ で代替表示されます（日本語キーボードにそのように刻印されています）。本書を通じてバックスラッシュは ‘\’ と表記しますが、読者は適宜読み替えてください。

図のソースファイルは次のようです（以下で... と表している箇所には好きなだけ、文章を挿入して下さい）

図 1.1 Windows/Mac 用 L^AT_EX 統合ソフトウェア TeXWorks の組版ボタンはウィンドウ左上にある矢印のついた緑色丸ボタン。組版エンジンとして”pLaTeX”を選択している。

い。何行にもわたっていくら長くても構いません)。

```
my_first_document.tex
1 \documentclass{jsarticle}
2 \title{サルカニ合戦顛末記}
3 \author{足柄金太郎}
4 \date{昔々}
5 \begin{document}
6 \maketitle
7 \tableofcontents
8
9 \section{コトの発端}
10 カニが大好きなおにぎりをもって歩いていました。
11 サルが柿の種と交換しようと近寄ってきたのです。
12
13 \subsubsection{ずる賢いサル}
14 そのサルは言葉巧みにずるく振る舞うばかりで、自分で働かずに暮らしていたのです。
15
16 \subsection{サルにだまされた蟹}
```

図 1.2 Mac 用 L^AT_EX 統合ソフトウェア TeXShop の組版ボタンはウィンドウ左上にある。組版エンジンとして”pLaTeX”を選択している。

```
17 | カニが柿の種を丹精込めて育てたおかげで沢山の柿の実がなりました。  
18 | サルはカニの代わりに柿を取ってくれるというのですが、カニにはくれずに  
19 | サルだけが柿を食べるばかりです。  
20 |  
21 | \section{一計をこらすカニたち}  
22 | だまされたと知ったカニはサルをやっつけるために話し合いました。  
23 | \end{document}
```

組版ボタンを押して無事に組版できたでしょうか。正しく入力すべき箇所を間違えると処理にエラーが生じて組版に失敗します。その場合、訂正すべき箇所を見つけて修正して、再び組版を試みます。では、その組版結果をよく観測して下さい。さらに、引き続いてもう一度組版ボタンを押してみてください。組版結果が変化したはずですが、どこが変化したか気づきましたか。

演習 1.1 論文に相応しいテーマとその表題を考え、図 1.1 のようにしてその骨格になるようなソースファイルを作成し組版してみなさい。

1.2 文章の基本を確認する

ここで、文章の基本を確認しておきましょう。文章における最小単位は**文** (sentence) で、一文は句点 (‘。’) またはピリオド (‘.’) で終わります。作文教室では、文の連なりを一まとまりの**段落** (paragraph) として文章を組み立てるように学びました。段落始めではその行頭は**字下げ** (indent) されて文章を読みやすくするために欠くことのできない重要な視覚効果を提供し、文意の把握に大いに貢献します。このような段落最後の文を書き終わってから新たな段落を開始するための改行を**段落改行**とといいます。

原稿用紙で段落改行するためには次の段落行頭の字下げのために自ら 1 マスを空けるように書きます。L^AT_EX では原稿用紙の場合と違い、次の節 1.2.2 で説明するように、段落改行して次の段落行頭の字下げのための空白文字は入力する必要はありません。自動的に新しい段落であるように字下げ下げられて組版されます。

1.2.1 行と改行

ワープロや適切に設定されたテキストエディタでは入力文字がウィンドウ幅に達すると、原稿用紙上の行端に達したときと同じように、入力文字列はそこで折り返されて次の文字は次行の先頭に表示されます。この状況を**行の折り返し** (line wrapping) と呼びます。ウィンドウ幅を変化させると表示される行数が変化するのはこのためです。

一方、入力途中で `return` キーまたは `enter` キーが押されると、不可視文字である**改行文字** (`\n` で表すことにします) が入力され、その後に入力した文字が次行に表示されるようになります。この意図的な文字入力を**改行**とといいます。改行文字を 2 つ続けると、次の行頭に何も入力されないままに再び改行されるために、1 つの**空行** (empty line) となります。コンピュータでは、非改行文字の並びがあつて次の改行文字までの文字列を**論理行** (logical line) とといいます。

コンピュータで利用する文字は**文字コード**として規格化されています。字として認識される通常文字以外に、空白文字 (半角空白 \neq 全角空白) やタブ文字そして改行文字、それ以外にも接続装置に特別な動作をさせるために制御文字も定義されています。これらを通常文字と区別するためにしばしば**不可視文字**と呼ぶことがあります。コンピュータにおける**テキストファイル**とは、多数の文字コードが連なった 1 つの長大な文字列です。テキストファイルをなす 1 つの文字列には、改行文字がところどころに挿入されて論理行が順に並んでいるようになっています。テキストファイルの入力・編集作業において、半角空白および全角空白、改行文字、タブ文字を可視化し、論理行番号を表示できるテキストエディタの利用が推奨されるの、こうした事情があるからです。

1.2.2 L^AT_EX で段落のある文章を書く

L^AT_EX では、文字列に対して段落改行するために

- 1 つ以上の空行を挟むか、
- 段落を改めたい文末で**段落コマンド** `\par` を入力する

のいずれかで表します。段落改行をしない限り、連続した論理行は 1 つの段落内の文として組版されます。また、1 つ以上続く空行は 1 つの空行と同じ段落改行として扱われます。注意したいのは、段落改行後の行頭の

字下げ (indent) のために空白文字を入力する必要がないということです。字下げは L^AT_EX システムによって自動的に組版されます。

次の文は、『吾輩は猫である』(夏目漱石)の冒頭の一節です。3行目に段落改行するための空行があります。

```
1 吾輩わがはいは猫である。↵
2 名前はまだ無い。↵
3 ↵
4 どこで生れたかとうと見当けんとうがつかぬ。↵
5 何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。↵
6 吾輩はここで始めて人間というものを見た。↵
```

段落改行のために空行を挟まずとも、次のように段落コマンド ‘\par’ を使っても同じ組版を得ます。

```
1 吾輩わがはいは猫である。↵
2 名前はまだ無い。↵\par↵
3 どこで生れたかとうと見当けんとうがつかぬ。↵
4 何でも薄暗いじめじめした所でニャーニャー泣いていた事だけは記憶している。↵
5 吾輩はここで始めて人間というものを見た。↵
```

以降の L^AT_EX ソースの例示では、煩雑さを避けるために改行文字↵は明示せずに、論理行番号だけを添えることにします。

さて、詩歌のように文の途中でわざと行を改めて、次行を字下げせずに文を書くことがあります。このような改行操作を**強制改行 (break)**といい、組版上は歴然と区別されます。強制改行は段落改行とは違った意味合いや視覚的効果があるからです。L^AT_EX では次のようにコマンド ‘\’ または ‘\newline’ を使います。

```
1 \noindent
2 雨ニモマケズ\\
3 風ニモマケズ\nnewline
4 雪ニモ夏ノ暑サニモマケヌ\\
5 丈夫ナカラダヲモチ\nnewline
6 慾ハナク\\
7 .....
```

『雨ニモマケズ』(宮沢賢治)

‘\’ と ‘\newline’ は共に強制改行して次行を字下げせずに組版する同じコマンドです。ここでは1行目で、空行が挟まれたり段落コマンドの後で段落改行されるべき箇所でも段落冒頭の**字下げを抑制**するコマンド ‘\noindent’ を使っています。

L^AT_EX ソースファイルを書くお勧めは、1文を1論理行で書いて段落改行するときには空行を挟んで段落の切れ目とするスタイルです。1つの文を書いて句点を入力したら↵キーで改行文字を入力して次行に次の文を書き綴るのです。このように記述すると後からソースファイルの編集・推敲作業がはかどるからです。テキストエディタに習熟することはパソコンのたいへん重要なスキルで、作業を劇的に効率化します。エディタでま

ず覚えるべきキー操作は、保存、コピー、貼りつけですが、次の段階はテキストカーソルの行頭・行末への移動です。

演習 1.2 論文に相応しいテーマについて、すこし書き進めて、未完成であっても構わないので、段落の冒頭を思いつくままに書きならべて組版してみなさい。

1.3 特別な文字の取り扱い

L^AT_EX で作文を書く基本は以上で尽きますが、ソースファイルに入力する文字そのものについて注意しておきましょう。

L^AT_EX では特殊な役割をする特別な記号文字が定められており、これを**特殊文字**と呼んでおり取り扱いには注意が必要です。また、入力した文字とは異なるように組版される文字も存在します。組版でエラーが発生したりまたは思うようにならない場合には、これらの特別な文字の取り扱い方を探ってみて下さい。

L^AT_EX における特殊記号を表 1.1 に掲げました。これらをその記号自体として L^AT_EX 文書として組版するためには、表のようにバックスラッシュ記号 ‘\’ をつけて特殊記号の意味をエスケープさせるか、対応するコマンドを使います。

まず最初に気をつけたい文字がパーセント記号 ‘%’ です。ソースファイルに直接にパーセント記号を書く、論理行末までの文字列を組版処理の対象としません。次を見てみましょう。

- | | |
|---|--|
| 1 | 65 歳人口の総人口に占める割合は、昭和 60 年に 10 %を超え、20 年後の平成 17 年には 20 %を |
| 2 | 超え、その 8 年後の平成 25 年に 25 %となって初めて 4 人に 1 人が高齢者となった。 |

このソースを組版すると、“高齢者人口の総人口に占める割合は、昭和 60 年に 10 超え、その 8 年後の平成 25 年に 25” となります。1 行目および 2 行目にある最初の記号 ‘%’ から行末までが組版されていません。もしもこの事実は好都合です。ソースコード内に**コメント**を書くために利用できるからです。文章のメモや推敲時に入力文字をむやみに削除してしまうのではなく、コメントアウトして残しておくことが可能になります。

文字	読み	役割	文字の記述法
%	パーセント	コメント開始	\%
\	バックスラッシュ	コマンドの開始	\textbackslash
{	左ブレース	グループ・引数開始	\{ または \textbraceleft
}	右ブレース	グループ・引数終了	\} または \textbraceright
\$	ドル	文中数式の開始・終了	\\$ または \textdollar
&	アンパサンド	表内のセル切り	\&
#	シャープ	マクロ定義の引数	\#
^	キャレット	数式上添字	\^ または \textasciicircum
_	アンダースコア	数式下添字	_ または \textunderscore
~	チルド	行分割しない空白	\~ \textasciitilde

表 1.1 L^AT_EX の特殊文字

表 1.2 には、 \LaTeX のソースファイル内の見た目の文字とは違うように組版される文字を掲げました。ただし、数式モードではそのままの文字として使うことができます。

文字	読み	表示	記述
<	より小さい	\textless	<code>\textless</code>
>	より大きい	\textgreater	<code>\textgreater</code>
	縦棒	\textbar	<code>\textbar</code>
"	二重引用符	\textquoteright	
`	アクセント	\textasciitilde	
'	引用符	\textquotesingle	

表 1.2 \LaTeX ソースファイルの見た目と異なる組版になる文字。ただし、数式中ではそのまま文字として使える。

演習 1.3 論文に相応しいテーマについて、文書構想メモや思いついた単語をコメントなどに加えて、組版してみなさい。

第 I 部

文書全体についての考え方

第 2 章

文書を構成する 文章 \neq 文書

文書であるための、最低限の要素はタイトル（題名）、著者名、日付という表題情報を文書の冒頭に掲げることです。さらに、本や長い報告書などでは見出しに加えて、見出しとその登場ページ数が対応した目次が備わっていればその活用価値が一層高まります。目次をどのように作成し、どのように記載すればよいのかを考えてみると、複雑で面倒な作業であることに気づきます。L^AT_EX ではこれらをきわめて簡単に、しかも自動的に実現します。

L^AT_EX ソースファイルにおいて、冒頭の文書クラスを宣言している行 '`\documentclass{...}`' と本文を開始する行 '`\begin{document}`' とで挟まれた部分を**プリアンブル部 (preamble)** といいます。

2.1 表題の出力

文書には**表題情報**がはっきりわかるように組版されていることが必要です。表題情報はタイトル、著者情報（所属などを含む）、日付の 3 要素からなり、それぞれコマンド '`\title{...}`', '`\author{...}`', '`\date{...}`' で指定し、それらを本文冒頭に書くコマンド '`\maketitle`' によって組版されます。3 つの表題要素はプリアンブル部で指定し、それらの順番は違っていても構いません。たとえば第 1 章で紹介したように、たとえば次のように書きます。

```
1 \documentclass{jsarticle}
2 \title{サルかに合戦顛末記}
3 \author{足柄金太郎}
4 \date{昔々}
5 \begin{document}
6 \maketitle % 表題情報の出力
7 .....
8 \end{document}
```

表題情報を表示するためには次のようになります。

- 表題要素がプリアンブル部に記述してあっても本文冒頭に '`\maketitle`' の 1 行がなければ表題情報は組版されない。
- 表題要素はタイトル、著者、日付の順に組版される。

- 表題要素 ‘\title{...}’ を省略して ‘\maketitle’ を書くと組版エラーになる。
- ‘\date{\today}’ とすると組版時の日付になる。表題の’ 日付は原則的には 2017/4/11 や 11 Apr. 2017 のような略記はしません。文化習慣や言語によって 4/11/29 は 2029 年 4 月 11 日や昭和 29 年 11 月 4 日などと誤解を招くためです。2029 年 4 月 11 日あるいは平成 29 年 4 月 11 日と曖昧さなく記述します。日付は、校了時や文書提出時または提出期限日のいずれかとするのが一般的です。

2.2 見出しを付ける = 構造化

文書構造とは、文書全体を大まかなブロックに分け、さらに各ブロックをより下位の小さなブロックに分け、というようにして、文書を階層的に分節化することで得られる文章構成全体の様子です。たとえば、第 1 章、第 1 章第 1 節、第 1 章第 2 節、…のような見出しがある書籍の構成はその典型です。番号付けられた見出しを備え、それらが目次で一覽できる文書は文書の理解と利用にとって本質的な役割を果たしています。

L^AT_EX では文書を分節化するためのコマンドが用意され、それらは同時に**見出し** (head line) を付ける役目も担っています。重要なことですが、見出しには章や節番号、項 (小節) 番号などの見出し番号を書く必要はありません。L^AT_EX システムが文節レベルに応じた見出し番号を自動的に付けて組版し、視覚的にも読みやすい文書を作成することができます。

文書構造の最上位レベルはソース冒頭の ‘\documentclass{...}’ で宣言されている文書クラスが定めています (第 2.4 節)。jsarticle では、最上位の文書構造レベルとして節 (section) を持ち、下位に向けて小節 (subsection)、小々節 (subsubsection) と文節化できるようになっています。それに応じて、節の見出し指定をコマンド ‘\section{...}’ で、小節の見出し指定を ‘\subsection{...}’ で、小々節見出し指定を ‘\subsubsection{...}’ で次のように指定します。

```

1 \documentclass{jsarticle}
2 \title{サルカニ合戦顛末記}
3 \author{足柄金太郎}
4 \date{昔々}
5 \begin{document}
6 \maketitle
7 \tableofcontents % 目次の生成
8
9 \section{コトの発端}
10 ....
11 \subsubsection{ずる賢いサル}
12 ....
13 \subsection{サルにだまされた蟹}
14 ....
15 \section{一計をこらすカニたち}
16 ....
17 \section{サルを奇襲する}
18 ....

```

```
19 \section{カニの凱旋}
20 ....
21 \end{document}
```

2.3 目次を作成する

上の例では見出しコマンドだけでなく、さらに7行目にコマンド“`\tableofcontents`”を追加しています。たったこれだけで、組版処理を2回以上繰り返すと見出しとその登場ページの対応が正しい目次が自動的に組版されます (2.5.2 節 (14 ページ) 参照)。

演習 2.1 論文に相応しいテーマを定めて現時点で構想した見出しを作成し (書いてみたい、あるいは書くべきことを列挙してみる)、文書全体を俯瞰できるように併せて目次も組版しなさい。

2.4 文書構造と文書クラス

L^AT_EX では文書全体を通じて首尾一貫した組版が自動生成できるため、文書の作成者は文書構造を明確にして解りやすく正確な文章の作成に集中することができます。L^AT_EX では文構造コマンドを使って文章の関係骨子を明確に組み立てながら、立体的な構造を持たせて書くことが可能です。文書を作成する場合、次のような過程を経て文書を作成することを考えてみます。

1. 文書の全体を構想し、見出し用コマンドを使って見出しを列挙する (目次構成) (14 ページ)。
2. 目次出力を通じて見出しを階層的にグループ化し、併せてそれぞれの構造に応じて文章を書き足す。
3. 文書全体の構成とバランスを考慮しながら、文章の推敲や文構成の入れ替え・削除などの編集を行なう。
4. 文章表現やレイアウトなどを調整して文書として完成する。
5. 必要に応じて、文書作成作業の中で索引用インデックをつけて、文書索引を生成する (50 ページ)。

2.4.1 文書構造の階層

1つの文書は階層的な内部構造のある文章だと見なすことができます。図 2.1 のように、文書の最上位レベルの文構造の単位として**部** (part) を持たせたとします。文書はそれから下位に向かって**章** (chapter)、**節** (section)、さらに**項** (subsection)、そして**目** (subsubsection) の構造単位に細分されていきます。また、さらに細かく**段** (paragraph) と**小段** (subparagraph) という単位を考えることもあります。図 2.1 には、階層的な構造単位として見出しを指定するための**文構造コマンド**も示しました。

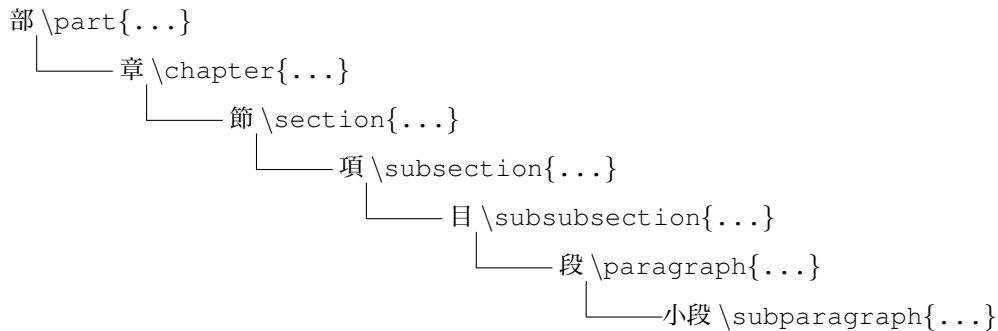


図 2.1
文書の階層構造とその見出し生成の L^AT_EX コマンド。最上位階層が、書籍クラスでは部 (`\part{...}`) または章 (`\chapter{...}`) から、論文クラスでは節 (`\section{...}`) から始まるように指定する。

2.4.2 文書クラス

文書によっては階層レベルの最上位が章 (chapter) であったり、あるいは節 (section) であることがあります。L^AT_EX では、文書の最上位階層は**文書クラス** (document class) が定めており、冒頭に置く

```
\documentclass{文書クラス名}
```

を使って宣言します。作成する文書の目的や性質に応じて、L^AT_EX システムには適切に組版される文書クラスがあらかじめ幾つか用意されています。

本書で主に利用する**日本語論文クラス**を使うには `jsarticle` を指定するのでしたね。論文クラスでは、図 2.1 において節 (`\section{...}`) を最上位の階層とするように、`\section{...}`、`\subsection{...}`、`\subsubsection{...}` と見出しを付けて書いていきます。論文クラスでは最上位レベルとして部 (`\part{...}`) や章 (`\chapter{...}`) は利用できません。

日本語書籍クラスを使うときには `jsbook` を指定します。書籍クラスでは、図 2.1 において部 (`\part{...}`) あるいは章 (`\chapter{...}`) が最上位階層であるように、たとえば、`\chapter{...}`、`\section{...}`、`\subsection{...}`、`\subsubsection{...}` と見出しを付けて書いていきます。

必ずしも、最上位の階層から下位のすべてにわたるように見出しを付ける必要はありません。ただし、文書クラスが許している最上位階層より下位の階層を見出しとするように文書を始めたり、たとえば `\section{...}`、`\subsubsection{...}` のように階層を飛ばして見出しをつけると、見出し番号の並びにゼロが入ることになります。大きな文構造 (深い文書階層) を持つ文書であっても、文書量が多いわけではありません。論文クラスであっても数百ページの論文を書くことも、書籍クラスを使って数十ページの本を書くことも問題ありません。

文書の最上位階層の定め方や階層構造の深さと文書量 (ページ数) とは対応関係はなく、文書階層はあくまでも文書を分節化するための印に過ぎません。実際には文書クラスは組版されて製本された読み物としての体裁によって選ばれる場合が多いようです。長大な文書だからといって必ずしも深い文章階層を持つ文書クラスを使用しなければならないわけではありません。

指定した文書クラスに応じて適切に見出しコマンドを使うと、組版の際に階層レベルに応じて見出しの登場順に自動的に通し番号が付きます。**見出し番号**は、文の階層レベルをピリオドで区切った数字の並びで表されます。たとえば、日本語論文クラス `jsarticle` の見出し書式は *n.m.k* で、*n* が節番号、*m* が項番号、*k* が目

番号とする番号となります。L^AT_EX には、文構造の階層レベルを表す見出し番号だけでなく、表番号や図番号を連番で組版する table および figure 環境も用意されています (4.4.1 節)。

文構造を指定する文構造コマンドは次のように ‘*’ を付けることができます。

```
\part*{...}           \chapter*{...}       \section*{...}
\subsection*{...}    \subsubsection*{...} \paragraph*{...}
\subparagraph*{...}
```

*付き文構造コマンドを使った場合、組版すると見出しは表示されますが見出し番号は付きません。また、これらの見出しは目次にも現れません。

見出しコマンドの登場順序が変更、追加・駆除されると、L^AT_EX システムは組版のたびに自動的に見出し番号の付替えを行いません。目次における見出し番号の整合性は 2 回続けて組版することで保証されます。文書作成者は文章の構成だけに専念しながら文を組み立てればよいのです。

文書クラスは文書構造だけでなく、それに応じた文書レイアウトも定めています。L^AT_EX では、一度文書クラスを決めてしまえば最終的に組版結果を得るまでの処理は自動化され、活字の大きさやバランスなど“仕上がり”に関する調整を気に掛ける必要がありません。文書内容そのものの作成・編集だけに集中すればよいのです。L^AT_EX に関する十分な知識があれば、自分が使っている文書クラスを自由にカスタマイズすることが可能です (奥村 [1])。

学会誌に投稿する研究論文のために専用の文書クラスが公開されています。出版社が独自の文書スタイルファイルを作者に提供することもあり、従来のように出版のための諸過程 (とくに校正) が短縮され、速やかな誤植のない出版とそのためのコスト削減を同時に達成しています。

一方、文書クラスとは別に書式や体裁を美しく簡単に整えるために、数多くの **パッケージ** が公開されています。これらを利用して L^AT_EX の表現力をさらに高めることが可能です (節 8)。

2.5 文書作成の進め方

以下、日本語論文クラス jsarticle に従って文書作成する場合を例に、文書作成の様子を確認してみましょう。

2.5.1 文書の表題

文書には必ず表題を付けるようにします。2.1 節 (9 ページ) で紹介したように、L^AT_EX では **表題情報** (title) 出力のための 4 つのコマンド ‘\title{...}’、‘\author{...}’、‘\date{...}’ および ‘\maketitle’ を 1 組にして表題情報の組版を指定します。本文冒頭に書く ‘\maketitle’ を除いて、これらの表題要素を次のようにプリアンブルに記述します。

用途	書式
表題	\title{...}
著者情報	\author{...}
日付	\date{...}
表題情報の出力 (必須)	\maketitle

文書には表題が必要です。表題要素を指定すると表題としての文書情報がマークされることになり、文書情報処理の際に重宝します。仮に、表題を出力しない場合には ‘\maketitle’ の行をコメントアウト (行頭に % を書く) にすればいいだけです。ファイルの冒頭に作成した文書に関する簡単なメモと日付などをコメン

トしておくことも大切です。

2.5.2 目次の出力

L^AT_EX では、文の論理構造を指定する文構造コマンドの情報に基づいて見出しが登場するページ数を対応させて、自動的に**目次** (contents) を作成することができます。目次を作成するための L^AT_EX コマンドは ‘\tableofcontents’ です。

目次を出力するときは、‘\begin{document}’ と ‘\end{document}’ ではさまれた範囲で、目次を出力したい場所に 1 行

```
\tableofcontents
```

と書くだけです。この目次コマンドは、表題部の直後の本文に先立つ場所に記入するのが普通です。

目次情報の入った文書を組版するためには 2 回の組版処理が必要です。最初の組版処理で L^AT_EX システムはソースファイルから見出し情報を取り出し、目次情報を記載した目次ファイル (Table Of Contents ファイル) を拡張子 ‘.toc’ を付けて作成します。この段階では組版ファイルにはまた目次情報を含みません。続けてもう 1 度組版処理をすると、この目次ファイルを読み込んで目次情報が取り込まれた組版ファイルが生成されます。これが目次を出力するためにファイルを 2 回組版処理しなければならない理由です。文章構造を変更した場合も、正しく目次情報を整えるためには 2 回続けて組版処理が必要になります。

2.5.3 L^AT_EX ファイルのスケルトン

以上をまとめて、L^AT_EX 文書を作成する際のファイルの一般的様子を以下に示します。

% 2nd Sep. 2010 作成	『%』から行末までの文はコメントとなる
% 12 Sep. 2010 修正	作成した日付を入れる
% コンパイル時の注意や、文書の目的なども記載しておくとい	
\documentclass{jsarticle}% jsarticle の利用を推奨	文書クラスの指定
% プリアンブル部	本文中で使うコマンドなどの約束事などを記す
\usepackage{graphicx}	利用するパッケージを指定
\usepackage{amsmath,amssymb}	
\title{タイトル}	\title{ }, \author{ }, \date{ }, \maketitle
\author{著者名}	で 1 組と考える
\date{日付}	
....	
% プリアンブルはここまで	
\begin{document}	これ以下が本文の始まり
\maketitle	表題の出力
\tableofcontents	目次の作成。2 回以上 jlatex をかける
\begin{absrtact}	

```

論文の要約が必要なら、\begin{abstract}と\end{abstract}で
囲まれたこの部分に書く。やや小さな字で印刷される。
\end{abstract}

\section{節の題名}
節の内容を書きます。いくら長くても短くても構わない。
次に \section{..} タグが現れるまでの文がこの節に属する。

\subsection{項の題名}
必要なら副節を付けることもできる。
\subsection{..}は直前の\section の下位に属する。

\subsubsection{目の題名}
\subsubsection{..}は直前の\subsection の下位に属する。
%\subsubsection{目の題名}          文書構造の推敲跡をコメントで残す

\section{節の題名}
ここから新たな節が始まります。節に付けられる番号は
\section{..}が登場してくる順番にふられる。
\subsection{..}や\subsubsection{...}についても同じ。
.....

\section{節の題名}
以下、同様に文を書いていく。
.....
\end{document}          ここで本文が終わる

```

このように、 $\text{L}^{\text{T}}\text{E}^{\text{X}}$ では文書としての文構造コマンドによって該当する見出し部分マークすることで最高度に美しい整形出力を実現するシステムです。誰にでも読みやすくわかりやすい文章を書くための基本は、

- 文章の意味内容や文構造をよく考え
- 文構造コマンド $\text{\section{..}}$, $\text{\subsection{..}}$ や $\text{\subsubsection{..}}$ などとうまく活用し、
- 適切な見出しつけて

文章を作成し、

- 文書にふさわしいタイトルを考え
- 文書の構造を一望できる目次をつけて

文書を完成することです。この文書作成支援システムとしての L^AT_EX の特性を利用しながら文書を書くことは、的確な表現力を身に付けるための有効な訓練の一つとなるでしょう。

2.6 長大な文書作成ではファイルを分割する

L^AT_EX で作成する文書が長くなってくると、1つのファイルでの編集作業に支障をきたしたり、またエラーを取り除くための組版処理時間も長くなってしまいます。このようなときにはファイルを分割し、元となるファイル**ルートファイル**としてこれに分割したファイルを読み込むように書いて組版処理することができます。図 2.2 はその様子を表しています。ルートファイルとそこから読み込まれる分割ファイル群を合併すると、`\documentclass{...}` による文書クラスの宣言から始まり、`\end{document}` で終わるように L^AT_EX ファイルとして完全であることが必要です。

このようにファイル分割すれば、目的の箇所を記述した部分だけに組版を試みて組版エラーを取り除いたり、文書内容ごとにファイル作成することで編集作業を容易にするだけでなく、容易に文書構造の変更が可能となります。とくに書籍や長大な論文レポートの効率的作成のためには、文書ファイルの分割とそれにかかわるファイル管理が鍵となります。

図 2.2 ルートファイル `my_article.tex` から分割ファイルを読み込んで文書を作成する。

分割したファイルを読み込むには、コマンド `\input` または `\include` で次のようにしてファイルを指定します。ファイル名はルートファイルから目的ファイルまでのファイルパスを含んでいます。パスの区切り記号は、Windows/macOS や Linux であっても共通で、`/` を使います。分割保存されるファイルの拡張子は L^AT_EX 原稿であることがわかるように拡張子 `.tex` を付けます。

```
\input ファイル名
```

または

```
\include ファイル名
```

`\input` と `\include` との違いは、`\include` 自身が強制的に改ページ (`\newpage`) を行って読み込んだ

ファイル内容を組版していくことです。したがって、読み込まれるファイルの先頭が改ページされて組版される場合、たとえば書籍クラスを使って`\chapter{...}`（章）で始まる見出しを持つような分割ファイルに対しては`\include`を使うと都合がよくなります。

図 2.2 では、次のように組版対象のルートファイル `my_article.tex` において`\input` を使って分割ファイルを読み込んでいます。ルートファイル `my_article.tex` を組版処理するとき、`\input` によってファイル `preamble.tex`, `title.tex`, `introduction.tex`, ..., `conclusion.tex`, `bibliography.tex` が読み込まれ単純に接続されて組版されます。

```
my_article.tex
\documentclass{jsarticle}
\input preamble.tex
\input title.tex
\begin{document}
\maketitle
\input abstract.tex
\tableofcontents
\input introduction.tex
...
...
\input conclusion.tex
\input bibliography.tex
\end{document}
```

読み込まれるファイルの中でさらに`\input` で別のファイルを読み込んでも構いません。しかし、ファイルの読み込みをネストさせるとルートファイルからの依存関係がわかりにくくなるのでお勧めしません。

文書作成においてファイル分割すると、`\input` の前にコメント記号`%`を付ければ、そのファイルの読み込みがスキップされることになるので、文書作成の過程において指定したファイルだけを組版対象として組版エラーを取り除くことができます。また、`\input` を入れ替えることによって容易に文書構成の変更を試してみることが可能になります。作成する文書が長大であるときや文書量が増えると予想される場合には、ルートファイルをまず作成し、分割ファイルが文書を構成するパーツとなるように相応に考えた読み込まれるファイル名を指定した`\input` 行をコメントアウトして **L**A_TE_X ファイル群の概要を設計して、個々の分割ファイルを仕上げるように作業するとよいでしょう。ちょうど、文書の見出しを書き出して目次を作成する過程と呼応しています。

第 3 章

文書のレイアウト

文書は、それを印刷したときにその内容がわかりやすいように文字の大きさや配置を工夫することが必要です。これを文書の**レイアウト**といいます。

3.1 書式と環境

文章のある部分を一定の書式で記述したい場合があります。たとえば、指定した文章範囲を引用文だとわかるように記述するとか、中央に寄せて記述するとか、箇条書となるようになどです。このような目的のために、 \LaTeX では**環境** (environment) という方法を用意しています。

文のある部分を指定した環境下に置くには次のような書式で記述します。環境の中で書体を変えても、その影響は環境の外には及びません。

```
\begin{環境名}
  環境に支配される文章
\end{環境名}
```

以下に、 \LaTeX 環境のいくつかを紹介します。

3.2 文書の概要

3.3 文の引用

文章中に ‘他の文章’ や誰かの ‘発言’ などをそのまま借用するために、その文の両端のマージンを余分にとってレイアウトするとわかりやすいことがあります。このような文章の表記を**文の引用**といいます。 \LaTeX には `quote` と `quotation` の二つの引用環境があります。

3.3.1 quote 環境

quote 環境 中の文章は、文体が字下げされます。段落の切れ目（空白行または `\par` で表す）では “段落の最初の文字の字下げが行なわれません”。また、**quote 環境** の前後にはやや広めの空白が確保されます。たとえば、次のような言葉を引用してみましよう。

```
\begin{quote}
さるかに合戦の勃発についての歴史上の驚異を探查するための我々の現在の方法の多くを提供したのは猿蟹大学の浦島田太郎の洞察であった。
実際、彼は合戦をその時代における歴史システムのダイナミックスとして理解することの重要性を強調したのであった。

彼の多くの示唆はその後、他の人により精密化され拡張されたが、我々が彼の創造力と洞察に負っていることは誇張し過ぎることはない。

\hfill 花咲爺、『さるかに合戦の考古学』
\end{quote}
```

このとき出力は次のようになります。

さるかに合戦の勃発についての歴史上の驚異を探查するための我々の現在の方法の多くを提供したのは猿蟹大学の浦島田太郎の洞察であった。実際、彼は合戦をその時代における歴史システムのダイナミックスとして理解することの重要性を強調したのであった。

彼の多くの示唆はその後、他の人により精密化され拡張されたが、我々が彼の創造力と洞察に負っていることは誇張し過ぎることはない。

花咲爺、『さるかに合戦の考古学』

3.3.2 quotation 環境

quotation 環境は `quote` 環境とほぼ同じです。ただし `quote` 環境と違って、段落の切れ目（空白行または `\par` で表す）では、通常文と同じように“字下げ”によって段落の切れ目が表わされます。たとえば、上の文例を `quotation` 環境で使うと出力は次のようになります。

さるかに合戦の勃発についての歴史上の驚異を探查するための我々の現在の方法の多くを提供したのは猿蟹大学の浦島田太郎の洞察であった。実際、彼は合戦をその時代における歴史システムのダイナミックスとして理解することの重要性を強調したのであった。

彼の多くの示唆はその後、他の人により精密化され拡張されたが、我々が彼の創造力と洞察に負っていることは誇張し過ぎることはない。

花咲爺、『さるかに合戦の考古学』

3.4 文を寄せる

文章を揃えて左寄せにしたり、中央寄せや右寄せに配置したい場合があります。このようなときには、それぞれ **flushleft 環境**、**center 環境**そして **flushright 環境**を使います。文を寄せる際には、**強制改行**コマンド `\\` を利用することがあります。

強制改行の場合には、改行後の行頭の文字の字下げは行なわれません。??節で説明したように、 \LaTeX では空白行または `\par` は段落の切れ目となり、改行された上で新段落の行頭文字が字下げされます。

center 環境 改行で区切られた文や指定した図表を中心に配置することを**中寄せ**といいます。

flushleft 環境 改行で区切られた文や指定した図表を左に配置することを**左寄せ**といいます。

flushright 環境 改行で区切られた文や指定した図表を右に配置することを**右寄せ**といいます。

これらの文を寄せる環境を使って次のように書いてみます。

```
\begin{flushright}
\LaTeX{}は世界中で\\
利用されている\\
文書整形の\\
定番です
\end{flushright}
\begin{center}
誰にでもできる\\
簡単な\\
入門
\end{center}
\begin{flushleft}
Linux、\\
Macintosh や\\
Windows でも OK です
\end{flushleft}
```

このとき出力は次のようになります。

\LaTeX は世界中で
利用されている
文書整形の
定番です

誰にでもできる
簡単な
入門

Linux、
Macintosh や
Windows でも OK です

3.5 擬似タイプ入力

コンピュータプログラムを掲載したり、文書の一部をタイプしたときと同じように出力したいときがあります。また、 \LaTeX コマンドを説明する場合には、特殊記号を含んだ \LaTeX ファイルの生原稿を出力する必要があります。このような目的のために、生原稿を**そのまま出力**する環境として **verbatim 環境**を使います。たとえば、本節の文頭の生原稿を出力するには次のように書きます。

```
\begin{verbatim}
\subsection{擬似タイプ入力}
....
\LaTeX{}コマンドを説明する場合には、特殊記号を含んだ\LaTeX{}ファイルの生原稿を出力する
必要があります。
このような目的のために、生原稿を\textbf{そのまま出力}する環境として
\textbf{verbatim 環境}....
\end{verbatim}
```

このとき出力は次のようになります。

```
\begin{verbatim}
\subsection{\texttt{verbatim}環境}
\LaTeX{}コマンドを説明する場合には、特殊記号を含んだ\LaTeX{}ファイルの生原稿を出力する必要があります。
また、ソースプログラムを掲載することも必要になるでしょう。
\end{verbatim}
```

そのまま出力したい‘生原稿部分’を文中で表すには、‘`\verb`’コマンドを使います。`\verb`の直後の文字(たとえば‘|’や‘+’)が次に表れるまでに囲まれた文字列が、たとえそれが \LaTeX コマンドであろうとも本来の意味機能を停止して、そのまま出力されます。たとえば

```
\verb|\LaTeX{}|文書は美しい
```

と書けば、‘`\LaTeX{}`文書は美しい’と出力されます。

環境名において `verbatim*`、または文中で ‘`\verb*`’ のように * を加えたときには指定された範囲の生原稿がそのまま出力されるのは同じですが、半角空白文字が ‘`␣`’ と表されるので使い分けると便利でしょう。

3.6 箇条書

箇条書をするための環境として **itemize 環境**、**enumerate 環境**および **description 環境**の3つがよく利用されており、それぞれ**単純箇条書**、**列挙箇条書**、**見出し付箇条書**と呼びます。

3.6.1 itemize 環境

箇条項目の先頭に ‘\item_’ をつけて箇条書きします (\item の後に空白 ‘_’, または改行が必要です)。単純箇条書では、各箇条項目の前に印 ‘•’ が付きます。単純箇条書は箇条項目の順番を入れ替えても問題が起こらないときに使います。箇条項目の順番に意味があるときには列挙箇条書を採用すべきです。

- itemize 環境では箇条項目に • が付きます。
- enumerate 環境では箇条項目に番号が付きます。
- description 環境では箇条項目に見出しを付けることができます。

と出力するためには、次のように書きます。

```
\begin{itemize}
\item itemize 環境では箇条項目に$\\bullet$ が付きます。
\item enumerate 環境では箇条項目に番号が付きます。
\item description 環境では箇条項目に見出しをつけることができます。
\end{itemize}
```

3.6.2 enumerate 環境

列挙箇条書環境でも、単純箇条書とまったく同じように箇条項目に先立って ‘\item_’ を付けます (\item の後に空白 ‘_’, または改行が必要です)。列挙箇条書では、各箇条項目の前に列挙した順に番号が振られます。

箇条項目の順番を入れ替えても問題がない場合には単純箇条書にすべきです。単純箇条書にするか列挙箇条書にするかは、その理由を考えてから適切な箇条書環境を選んでください。

1. itemize 環境では箇条項目に • が付きます。
2. enumerate 環境では箇条項目に番号が付きます。
3. description 環境では箇条項目に見出しをつけることができます。

と順序数を付けて出力するには、次のように書きます。

```
\begin{enumerate}
\item itemize 環境では箇条項目に$\\bullet$ が付きます。
\item enumerate 環境では箇条項目に番号が付きます。
\item description 環境では箇条項目に見出しをつけることができます。
\end{enumerate}
```

3.6.3 description 環境

箇条項目に先立って ‘\item[...]’ を付けて [...] 内に箇条項目の見出しを書いて、見出し付き箇条書とします。

単純箇条書 itemize 環境では箇条項目に ● が付きます。

列挙箇条書 enumerate 環境では箇条項目に番号が付きます。

見出し付き箇条書 description 環境では箇条項目に見出しを付けることができます。

と出力するには、次のように書きます。

```
\begin{description}
\item[単純箇条書] itemize 環境では箇条項目に $\bullet$ が付きます。
\item[列挙箇条書] enumerate 環境では箇条項目に番号が付きます。
\item[見出し付き箇条書] description 環境では箇条項目に見出しを付けることができます。
\end{description}
```

3.7 ネストされた環境

¥ LaTeX の環境では、次のように環境内の文の中に環境をネスト、つまり環境文を多重化させることができます。

```
\begin{環境 A}
.....
\begin{環境 B}
.....
\begin{環境 D}
.....
\end{環境 D}
.....
\end{環境 B}
.....
\begin{環境 C}
.....
\end{環境 C}
\end{環境 A}
```

上の例では、環境 A の文書の中に環境 B と環境 C が使われています。さらに、環境 B の中には環境 D が使われています。このように環境の中でさらに環境を使うことを**環境のネスト**といいます。

ネストさせる環境は、原則としてどんなものでも構いません。たとえば、単純箇条書の環境を次のようにネストさせてみましょう。

```
\begin{itemize}
\item 単純箇条書第 1 レベル
  \begin{itemize}
\item 単純箇条書第 2 レベル
  \begin{itemize}
\item 単純箇条書第 3 レベル
  \begin{itemize}
\item 単純箇条書第 4 レベル
  \end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}
\item 単純箇条書第 2 レベル
\end{itemize}
```

すると、次の出力結果が得られます。

- 単純箇条書第 1 レベル
 - 単純箇条書第 2 レベル
 - * 単純箇条書第 3 レベル
 - ・ 単純箇条書第 4 レベル
- 単純箇条書第 2 レベル

同様に、列挙箇条書の環境を次のようにネストさせてみましょう。

```
\begin{enumerate}
\item 列挙箇条書第 1 レベル
  \begin{enumerate}
\item 列挙箇条書第 2 レベル
  \begin{enumerate}
\item 列挙箇条書第 3 レベル
  \begin{enumerate}
\item 列挙箇条書第 4 レベル
  \end{enumerate}
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

```
\end{enumerate}
\item  列举箇条書第 2 レベル
\end{enumerate}
```

すると、次の出力結果が得られます。

1. 列举箇条書第 1 レベル
 - (a) 列举箇条書第 2 レベル
 - i. 列举箇条書第 3 レベル
 - A. 列举箇条書第 4 レベル
2. 列举箇条書第 2 レベル

これらの箇条書きを入れ子にしたときに、その深さに応じて自動的につけられる項目記号は再定義することができます（奥村 [1, 第 5 章箇条書き]）。

いままでの例からもわかるように、 \LaTeX は文のレイアウトや後で述べるように文書クラスを定めることによって最高度に美しい整形出力を実現するシステムです。最初からエラーのない文書を書くことは誰にもできません。 \LaTeX システムでのコンパイルとエラーの修正を繰り返しながら徐々に完成していくのが一般的な過程です。そのために、できる限りエラーの見つけやすい文書書法を心掛けましょう。 \LaTeX ファイルでは、その印刷結果が同じであってもかなり自由に書くことができます。プログラム言語を記述するときと同じ態度ですが、字下げ等を使って文書の論理構造を明示する書き方を踏襲するなどの工夫をするとよいでしょう。

3.8 脚注と傍注

\LaTeX では、簡単に**脚注**を付けることができます脚注には文中で

```
\footnote{..}
```

を使って `{..}` 内に脚注を記述します。冒頭に `\label{ラベル名}` によってラベルを付けて、文中の任意の場所で `\ref{ラベル名}` によって脚注番号を取得できます。脚注をイタズラに多用すると読み手の視線が散乱して読むリズムが狂うことがあるので注しましょう。脚注^{*1}では、本文の該当箇所に脚注を表す印が付けられ、同一ページ下のフッタの中に脚注文が表示されます。ここで書いた脚注番号^{*1}の脚注は次のように書きました。

```
脚注
\footnote{
\label{explanate_footnote}
これが脚注です。あまり多用すると文章が読みづらくなります。
}
では、本文に脚注を表す印が付けられ、同一ページ下のフッタの中に脚注文が表示されます。
```

*1 これが脚注です。あまり多用すると文章が読みづらくなります。

3.9 L^AT_EX で使われる文字

3.9.1 基本文字サイズ

出版の世界では、文字の大きさを**ポイント**という単位で表すことがあります。これは1インチを約72ポイントと考えて文字の大きさを計るやり方です。

L^AT_EX 文書の印刷仕上がりは、特に指示をしない場合には欧文文字が10ポイントの大きさに印刷されます。これでは見にくいと感じるならば、欧文文字を11ポイント、または12ポイントで印刷するオプションを指定することができます。このためにはファイル先頭の `\documentclass{...}` の箇所ですべてのようにオプションを指定します。

11ポイント	<code>\documentclass[11pt]{...}</code>
12ポイント	<code>\documentclass[12pt]{...}</code>

3.9.2 フォントの種類

印刷の世界では同一種類の活字1セットを**フォント** (Font) といいます。文書として効果的な出力結果を得るために、フォントの大きさや種類を変えることは出版の世界で広く行なわれてきました。L^AT_EX でもフォントの種類やその大きさを変更することができます。ただし、フォントを変更しなければならない理由をよく考えるべきで、むやみに多くのフォントを多用すると読みづらな結果になってしまうことがあります。

3.9.3 日本語フォント

標準的に利用できる L^AT_EX の日本語フォントは、原則として**明朝体**族と**ゴシック体**族の2書体です。特に指定しなければ、日本語全角文字は明朝体で印刷されます。文書中の日本語文字列を**ゴシック体** (Gothic) に変更するには、次のように指定します。

この部分が`\textgt{ゴシック体}`になる

あるいはボールド体 (太字) を指定する `\textbf{...}` を使っても同じ効果が得られます (論理的には、この `\textbf{...}` を使うほうがベターです)。

3.9.4 英文フォント

L^AT_EX で利用できる英文書体には次の表 3.1 にあるフォントがあります^{*2}。文書中の英文文字列のフォントを変更するには表 3.1 のようにして文字列範囲を指定します。

^{*2} jsarticle、や jbook などの文書クラスを指定した場合、数式で使われるフォントとしては通常 Computer Modern が使われます。

出力	入力	出力	入力
Roman	<code>\testrm{Roman}</code>	Boldface	<code>\textbf{Boldface}</code>
<i>Emphasis</i>	<code>\emph{Emphasis}</code>	Sans Serif	<code>\textsf{Sans Serif}</code>
<i>Italic</i>	<code>\textit{Italic}</code>	typewriter	<code>\texttt{typewriter}</code>
SMALL CAPS	<code>\textsc{Small Caps}</code>		

表 3.1 英文フォントを指定するコマンド

3.10 シンボル・特殊記号の表現

TeX システムは本来、数式が入り混じった文書の組版を目指して D. Knuth によって開発されました。TeX ファミリーで組版される数式は、どんな複雑なものでも美しく出力されます。ここでは数式の取り扱いは述べませんが、奥村 [1] を参照してください。

L^AT_EX で用意されている豊富な記号を利用するためには**数式モード**内で記号指定をします。数式モードとは $\$$ で始まり $\$$ で閉じる特別な状態を意味しています。たとえば、♡を表すためには $\$\heartsuit$$ と記入します。

3.10.1 ギリシャ文字

ギリシャ文字を使う場合にも数式モード内で表 3.2 にあるように指定します。

出力	入力	出力	入力	出力	入力
α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>
δ	<code>\delta</code>	ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>
ζ	<code>\zeta</code>	η	<code>\eta</code>	θ	<code>\theta</code>
ϑ	<code>\vartheta</code>	ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>
ξ	<code>\xi</code>	o	<code>o</code>	π	<code>\pi</code>
ϖ	<code>\varpi</code>	ρ	<code>\rho</code>	ϱ	<code>\varrho</code>
σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>
υ	<code>\upsilon</code>	ϕ	<code>\phi</code>	φ	<code>\varphi</code>
χ	<code>\chi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>
Ψ	<code>\Psi</code>	Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>
Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>	Θ	<code>\Theta</code>
Π	<code>\Pi</code>	Φ	<code>\Phi</code>	\sum	<code>\sum</code>
\prod	<code>\prod</code>				

表 3.2 ギリシャ文字群

3.10.2 記号

ギリシャ文字以外にも、表 3.3 にあるように多くの記号やシンボルが数式モードで使うことができます。

出力	入力	出力	入力	出力	入力
±	<code>\pm</code>	∓	<code>\mp</code>	×	<code>\times</code>
÷	<code>\div</code>	*	<code>\ast</code>	★	<code>\star</code>
○	<code>\circ</code>	●	<code>\bullet</code>	⋅	<code>\cdot</code>
◇	<code>\diamond</code>	△	<code>\bigtriangleup</code>	▽	<code>\bigtriangledown</code>
◁	<code>\triangleleft</code>	▷	<code>\triangleright</code>	◯	<code>\bigcirc</code>
†	<code>\dagger</code>	≤	<code>\leq</code>	≥	<code>\geq</code>
≡	<code>\equiv</code>	~	<code>\sim</code>	≈	<code>\simeq</code>
≈	<code>\approx</code>	≠	<code>\neq</code>	α	<code>\propto</code>
⊥	<code>\perp</code>	∥	<code>\parallel</code>	←	<code>\leftarrow</code>
⇐	<code>\Leftarrow</code>	→	<code>\rightarrow</code>	⇒	<code>\Rightarrow</code>
↔	<code>\leftrightarrow</code>	↔	<code>\Leftrightarrow</code>	←	<code>\longleftarrow</code>
⇐	<code>\Longleftarrow</code>	→	<code>\longrightarrow</code>	⇒	<code>\Longrightarrow</code>
↑	<code>\uparrow</code>	↑	<code>\Uparrow</code>	↓	<code>\downarrow</code>
↓	<code>\Downarrow</code>	↕	<code>\updownarrow</code>	↕	<code>\Updownarrow</code>
↗	<code>\nearrow</code>	↘	<code>\searrow</code>	↙	<code>\swarrow</code>
↖	<code>\nwarrow</code>	∠	<code>\angle</code>	♭	<code>\flat</code>
♮	<code>\natural</code>	♯	<code>\sharp</code>	\	<code>\backslash</code>
∂	<code>\partial</code>	∞	<code>\infty</code>	△	<code>\triangle</code>
♣	<code>\clubsuit</code>	◇	<code>\diamondsuit</code>	♡	<code>\heartsuit</code>
♠	<code>\spadesuit</code>				

表 3.3 L^AT_EX で利用できる記号・シンボル (一部)

\mathcal{A} M \mathcal{S} -T_EX 用のスタイルファイル `amsmath` と `amssymb`^{*3} を使うためにプリアンブル部に

```
\usepackage{amsmath,amssymb}
```

と記述すると、さらに利用出来る数学記号がぐんと増加します。たとえば、ギリシャ文字 π (`\pi`) の太字を `\boldsymbol{\pi}` で $\boldsymbol{\pi}$ とできることや、不等号記号も `\leqq` で \leqq 、`\geqq` で \geqq 、`\lneqq` で \lneqq 、`\gneqq` で \gneqq などと多彩な記述が可能になります。数式を多用する文書を書くときには \mathcal{A} M \mathcal{S} -L^AT_EX の記述を使うとぐんと作業が楽になるため、調べてみる価値があります。たとえば、

```
\[
\cfrac{1}{\sqrt{2}}+ \cfrac{1}{\sqrt{2}}+
```

*3 \mathcal{A} M \mathcal{S} -L^AT_EX のスタイルファイルの入手には [TeXWiki/AMS-LaTeX](http://www.tug.org/texwiki/AMS-LaTeX) や本家のアメリカ数学会 <http://www.ams.org/publications/authors/tex/amslatex> から。

```
\cfrac{1}{\sqrt{2}+\dotsb }}
\]
```

と書くだけで、次のような連分数を書くことができる。

$$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}}$$

3.10.3 特殊文字とアクセント記号

特殊文字は表 3.4 のようにして出力します。‘ı’ や ‘ı̇’ のように、ドットのない英字を出力できることに注意してください。これらは、以下のように**アクセント記号**を組み合わせる利用します。またハイフン ‘-’ と 2 分ダッシュ ‘-’ と全角ダッシュ ‘—’ の使い分けにも留意してください。

出力	入力	出力	入力	出力	入力
%	\%	{	\{	}	\}
#	\#	\$	\\$	\$	\\$
^	\textasciicircum	~	\textasciitilde	_	_
\	\textbackslash	§	{\S}	¶	{\P}
©	{\copyright}	†	{\dag}	‡	{\ddag}
£	{\pounds}	¥	Y\llap=	œ	{\oe}
Œ	{\OE}	æ	{\ae}	Æ	{\AE}
å	{\aa}	Å	{\AA}	ø	{\o}
Ø	{\O}	ß	{\ss}	¿	?\`
ı	!\`	ı	{\i}	ı̇	{\j}
‘	\`	’	\`	“	\`
”	\`	-	-	-	--
—	---	L ^A T _E X	\LaTeX	T _E X	\TeX

表 3.4 特殊記号の印刷

フランス語などでは、通常のアルファベット以外にアクセント記号を組み合わせます。たとえば、`\c{c}` で cedilla を表し、`fa\c{c}ade` などと書きます。その他のアクセント記号を表 3.5 にまとめておきます。

名前	入力	出力	名前	入力	出力	名前	入力	出力
grave	\`{a}	à	tilde	\~{o}	õ	check	\v{s}	š
acute	\' {e}	é	bar	\={y}	ȳ	long	\H{j}	ĵ
hat	\^{o}	ô	dot	\. {p}	ṑ	tie-after	\t{\i u}	îu
umlaut	\" {u}	ü	breve	\u{\i}	ı̆	dot-under	\d{h}	ḥ

表 3.5 L^AT_EX のアクセント記号

第II部

それぞれのテクニック

第 4 章

正確な文書と相互参照

文書を作成する際に、参考情報を開示しその利用法を明らかにすることは文書作成における重要なマナーの 1 つです。文書作成のために参照した書籍・雑誌・新聞・Web 情報などの参考情報は、文書の本文中でその**出典**（出所情報）を**引用**し、同時にその参考情報の取り扱いを明らかにする必要があります。これらの参考情報は文書を成立させるための根拠情報です。参考情報の引用の際には、その書籍や論文や Web ページなどに確実に特定しアクセスできるために必要な著者。題名、出版社や発表年などを含む**書誌情報**をもれなく記さねばなりません。出典を明らかにし、本文中のどこで・どのようにそれを利用したかをその都度明示するための引用の工夫それ自体が読者の信頼を獲得し文書の正確さと価値を高めることとなります。

また、参考情報以外にも、同一文書内の別の箇所に言及しながら文書を書き進めることが期待されています。実際、文書のどの箇所（章や節番号またはページ数）で何を論じたか（あるいは論じようとしているか）を適宜示しながら論述する文書は読者の文書への注目を維持させ理解を助けます。図表を用いた文書では図表に番号を与えることで目的の図表を特定するように書くことで、‘上の図’や‘下の表’などのような曖昧な表現を排することができます。

正確な文書とは、文書量や参考情報や図表の数に依らず、文章作成の途中であっても常に正しく参考情報が引用され、言及箇所や図表番号が明らかにされている文書のことです。文書作成の任意の段階でも正しいことが大切で、文書作成の完成段階にだけこのような作業を行うことは非現実的であることに注意してください。作業終盤に文書構成の組み替えや追加・削除があったり、参考情報や図表の差し替え・挿入・削除があれば、すべてやり直さねばなりません。L^AT_EX では、常に正しく文書を書くための仕組みが整っています。

4.1 参考文献リストの作成

文書の作成に利用した参考情報は通常、文書の末尾に列挙します。L^AT_EX では参考情報は**参考文献**として、文献番号（名称）を添えて参考情報を列挙するように組版できます。これらの参考情報を文中で**文献番号**で参照することができます（文献番号は数字になるとは限りません）。

参考文献として組版するには専用の環境 **thebibliography** を使って次の書式にしたがって書きます。具体的様子を図 4.1 に示しました。パラメータ指定やオプションの使い方については以下に説明します）。

```
\begin{thebibliography}{字下げ幅パラメータ}
\bibitem[オプション番号 1]{引用ラベル名 1} 参考情報の出典.
\bibitem[オプション番号 2]{引用ラベル名 2} 参考情報の出典.
```

```
.....  
.....  
\end{thebibliography}
```

thebibliography 環境では、`\begin{thebibliography}{字下げ幅}` のように、参考情報が 2 行以上にわたるときの字下げ幅の目安となるパラメータを指定しなければなりません。文献番号を数字とすす通常の場合では、1 桁の文献項目数のときには ‘9’ を、2 桁になるときは ‘99’ のように桁数に合わせて指定すればよいでしょう。漢字幅とするとき、1 文字分なら ‘1zw’、2 文字分なら ‘2zw’ のようにします (図 4.1 参照)。

`\bibitem` の引数 ‘引用ラベル名’ は必須です。以下に説明するように、文書の任意の場所でコマンド `\cite{引用ラベル名}` を使うと対応する文献番号が組版され、その箇所で参考情報を文献番号として参照できます。引用ラベル名では大小文字は区別され、文書内で一意的であるように名付けねばなりません。

`\bibitem` のオプション ‘[オプション番号]’ を省略すると (図 4.1 の 11,12 行)、文献番号としてリストした順に [1], [2], [3], ... のように数字が付きます。文献番号が数字で組版されても混乱がないようであればオプションを省略します (通常はそうします)。オプションを使って `\bibitem[夏目 1905]{wagahai}`、`\bibitem[夏目 1906]{bocchan}` のように指定すると (図 4.1 の 13,14 行)、オプション番号として指定した文字列 ‘夏目 1905’、‘夏目 1906’ が文献番号として組版されます。文献番号を数字とするか名前と数字の組み合わせなどとするかは、投稿先の雑誌の指示や作成する文書が取り扱われる分野の慣例にしたがうとよいでしょう。

たとえば図 4.1 のように書くことができます。ただし、この例では文献番号が 11,12 行目では数字で 13,14 行目が指定した文字列で組版され、あえてスタイルを一貫させていません。実際の文献では文献番号の組版スタイルを統一するように書いて下さい。

本文の任意の箇所で参考情報の文献番号を参照するには次の書式にしたがってコマンド `\cite` を使います (図 4.1 参照)。

```
\cite[テキスト]{引用ラベル名}
```

文中で `\cite` のオプション ‘[テキスト]’ を省略して ‘`\cite{引用ラベル名}`’ とすると、対応する文献番号が組版されて参照できます。文書作成途中で資料を付け加えたり資料を並べ替えても、2 回続けて組版処理を行うと、参考文献の参照は破綻なく常に正しい文献番号となります。

文中で文献番号だけでなくページ数や章・節などを併せて組版したい場合はその文字列テキストをオプションで指定して `\cite[テキスト]{文献ラベル}` のように書くと、‘[文献番号, テキスト]’ のように組版されます。同じ参考情報の異なる箇所を引用する場合、異なる参考情報とせず同じ文献番号とし、オプションでその場所 (章・節やページ番号) を指定するようにします。

参考文献リストの参照が文書中に取り込まれるためには、やはり文書ファイルを最低 2 回コンパイルしなければなりません。aux ファイルに書き出された参考文献に関する情報を読み込むためです。

4.2 Web ページ情報の参照

参考文献としてインターネット情報を書誌情報として記載する場合には格別の注意が必要です。雑誌・新聞や書籍が紙媒体を経由せずにインターネット情報として流されるようになり、資料検索と収集の場は図書館か

```

1   ....
2   人社会の限りなく多くの苦難が「人権宣言集」\cite{human_rights}の一文一文に
3   刻まれている。
4   その最後に 1948 年 12 月の第 3 回国際連合総会で採択された世界人権宣言（人権に関する
5   世界宣言）が集録されている\cite[p.398]{human_rights}。
6   ....
7   伸びやかで平易な文章からなる「學問のすゝめ」\cite{fukuzawa}は、冒頭の余りにも
8   有名な一節から始まるように自由平等・独立自尊の思想、実学の奨励を当時の国民に
9   普及させた。
10  ....
11
12  \begin{thebibliography}{2zw}
13  \bibitem{human_rights} 高木八尺，末延三次，宮沢俊義編，『人権宣言集』，岩波文庫，1957.
14  \bibitem{fukuzawa} 福澤諭吉，『學問のすゝめ』，講談社学術文庫（2006）.
15  \bibitem[夏目漱石 1]{natsume1} 夏目漱石，『行人』，岩波文庫（1952）.
16  \bibitem[夏目漱石]{natsume2} 夏目漱石，『門』，新潮文庫（1986）.
17  .....
18  \end{thebibliography}

```

図 4.1 文書末の参考文献リストの記述と本文中の文献の引用

らインターネットに移行しつつあります。ニュース記事が配信され学術論文がオンラインで閲覧でき、各国政府や国連などから各種データをはじめ審議過程や報告文書も Web ページで公開されるようになり、オリジナル情報としてダウンロードできる場合も少なくありません。

インターネット情報へのアクセスにはそのために相応しいソフトウェア、たとえば Web ブラウザの利用が代表的です。Web 検索サービスから得られたリストをクリックしたり、目的情報の URL (Uniform Resource Locator) を直接入力して閲覧します。参照したインターネット情報の URL を正しく LaTeX ファイルに入力するためには、目的ページを表示させた Web ブラウザに示されている URL 文字列をすべて選択してコピーします。インターネット情報における書誌情報は、少なくとも次の内容を含む必要があります。

著者（団体名称）， 情報タイトル， 作成日付， URL

書誌情報は読者が確実に参考情報にアクセスすることを保証するためであることは言うまでもありません。ただし、もう一つ、私たち読者が多少手間をかけたとしても著者が参考にした情報を読者自身が参照すべきかどうかの材料を提供するという意味があります。その様子を具体的に確かめてみましょう。

たとえば、海洋の長期変化傾向について気象庁が発表している日本近海の海面水温に関する書誌情報を次の内容で参考文献にリストすることを考えてみましょう。

気象庁，「海面水温の長期変化傾向（日本近海）」（平成 28 年 3 月 10 日）、http://www.data.jma.go.jp/gmd/kaiyou/data/shindan/a_1/japan_warm/japan_warm.html

日本では最も信頼性が高いデータ提供元の1つである気象庁が著者（情報提供者）であること、ページタイトルと日付が明記されており参照情報がどのような内容であり何時のデータかも明確です。参考文献としてこれがリストされている文書を読んでいる（読もうとしている）読者に、この情報に目を通して見る価値がありそうだという判断材料とそのためアクセス方法を与えていると言えそうです。

4.2.1 URL の表記

ただし、この URL には L^AT_EX における特殊文字 ‘_’（アンダーバー）が含まれていることに注意して下さい。URL には他にも特殊文字 ‘&’ や ‘%’ が混在する場合があります。節 1.3 で L^AT_EX の特殊文字についての取り扱い（特殊文字の前にバックスラッシュ ‘\’ を付けて本来の役割をエスケープさせて組版するやり方）を紹介しました。しかし、URL についてこうした作業をわざわざ行うことは面倒な作業であり、しかもその途中で正しい URL 文字列を損ねてしまう可能性もあります。

URL の表記については次の例のように更に複雑な問題にも及んでいます。次のようにパーセント記号 ‘%’ が入り交じっている URL もしばしば登場します。次のような URL 文字列を入力して参考情報にアクセスしてその内容を確かめようとは誰も思わないのではないのでしょうか。

<https://ja.wikipedia.org/wiki/%E5%A4%8F%E7%9B%AE%E6%BC%B1%E7%9F%B3>

この例は、URL の表記において、英大小字、数字、‘!’（感嘆符）、‘(/ ’（丸かっこ）、‘_’（アンダーバー）、‘-’（ハイフン）、‘*’（アスタリスク）、‘.’（ピリオド）以外を使用できない文字を記号 ‘%’ を使って符号化することで目的の URL を表現しています。この方法を URL のパーセント符号化あるいは URL エンコーディングといい、インターネット技術の標準化文書 RFC3986 節 2.1 で定義されています。先の URL は <https://ja.wikipedia.org/wiki/夏目漱石> としてアクセスされる Wikipedia の Web 情報でした。この表記を復号化された URL ということにします。このとき、読者に知らせるべき Web 情報の望ましい書誌情報は復号化された URL を使って次のようになります（残念ながら Wikipedia には日付が見当たりません）。

Wikipedia, 「夏目漱石」, <https://ja.wikipedia.org/wiki/夏目漱石>

使用している OS や Web ブラウザによって URL の取り扱いは異なります。ブラウザの URL 表示をコピーして手コストエディタなどに貼りつけてもこのように URL が符号化されてしまうことがあります。何かの Wikipedia 情報にアクセス得てみて、その URL 表示を確かめたり、L^AT_EX やワードプロセッサのファイルに貼りつけてみてください。

パーセント符号化された URL を復号化して符号化前の復号化された URL を教えてくれる web サービスがあります。キーワード ‘パーセント符号化’ または ‘URL エンコード’ などで検索してみてください。URL はパーセント符号化された URL を正しく組版できたとしても、その URL を ‘読んでも’ 意味不明であり、しかも多くの場合は文字列は長大で URL を入力することは事実上不可能です。書誌情報として URL を文字列で表す場合には、復号化された URL を正しく表記するだけでなく、著者と Web ページタイトルも併せて記載するようにしてください。

4.2.2 パッケージ hyperref を使う

これまで言及することはありませんでしたが、これから先 L^AT_EX で組版作業する場合にはいつでも、以下のようにパッケージ hyperref を使うように宣言して下さい。宣言するだけで大変大きな効果が得られるからです。

インターネット情報を参考文献としてリストする場合、完全な書誌情報を載せることが確かめられました。コピーして穂林込んだ URL に含まれる特殊文字を正しく組版するためにさらに作業を重ねることは、ただしい URL を損ねてしまうことがあるためにお勧めしません。このために L^AT_EX システムに標準的に備わっているパッケージ hyperref を利用します。パッケージとは、L^AT_EX の機能を拡張したり新たな機能を追加したりするために誰でも使い方さえ知っていれば簡単に利用できる仕組みで、プログラム言語の世界ではライブラリあるいはモジュールと呼ばれているものです。

パッケージ hyperref の利用を宣言することは簡単で、プリアンブル部(ファイル冒頭の `\documentclass{文書クラス}`) と本文を開始する `\begin{document}` の間に次の 2 行を書くだけです。ここでは、L^AT_EX で組版して生成される目的の PDF ファイルを直接閲覧する際に、‘しおり’ (Adobe Acrobat Reader の用語) または ‘目次’ (macOS プレビューの用語) として利用するとき文字化けを防ぐためのパッケージ pxjahyper の使用も併せて宣言します (この行は hyperref の利用宣言よりも後になければなりません)^{*1}。hyperref のオプションとして [dvipdfmx] の指定は、L^AT_EX 組版処理でまず得られる DVI ファイル (DeVice Independent file) を PDF ファイルへと変換するプログラムとして dvipdfmx を使っていることを通知するためです (TeXWorks や TeXShop など大抵の L^AT_EX 統合環境での標準設定)。

```
\usepackage[dvipdfmx]{hyperref}
\usepackage{pxjahyper}
```

こうしてパッケージ hyperref と pxjahyper の利用を宣言するだけで他に何もしなくても、組版されて得られる PDF ファイルを PDF 閲覧ソフトウェアで開くと劇的な効果が生じていることがわかります。目次や本文中の文献番号、さらには相互参照によって組版した章・節番号や図表番号 (節 4.4 参照)、さらには索引 (節 7 参照) をクリックするとその場所にジャンプします。1 つのファイル内部の別の場所にジャンプするため、この紐付けをローカルリンクといいます。リンク先情報が埋め込まれた文字列をアンカー文字列と呼ぶのですが、目次や本文中の文献番号相互参照によって組版した章・節番号や図表番号および索引はパッケージ hyperref の利用すると自動的に色づけされたアンカー文字列となって目的の場所に関連付けられます。色づけなどの hyperref のデフォルト設定を変更するには次のようにプリアンブルで設定します。これらのパラメータの一部とその意味を表 4.1 に掲載しました。詳しい情報は hyperref^{*2} または TeX Wiki: hyperref^{*3} の Web ページにあります。

```
\usepackage[dvipdfmx]{hyperref}
\usepackage{pxjahyper}
```

^{*1} パッケージ pxjahyper(<https://www.ctan.org/tex-archive/language/japanese/pxjahyper>) は最近の TeXLive などの L^AT_EX 配布システムに標準パッケージとして含まれています。

^{*2} hyperref: Extensive support for hypertext in LaTeX, <http://mirrors.ibiblio.org/CTAN/help/Catalogue/entries/hyperref.html>

^{*3} TeX Wiki: hyperref, <https://texwiki.texjp.org/?hyperref>


```
\hypersetup{
  setpagesize=false,%
  bookmarksnumbered=true,%
  bookmarksopen=true,% しおりを開く
  colorlinks=true,
  linkcolor=blue,% 内部リンクを青
  citecolor=red,% 文献番号リンクを赤
}
```

オプション	値	意味・機能 (デフォルト値)
bookmarks	true/false	しおりを作成するか (true)
bookmarksnumbered	true/false	しおりに節番号をつけるか (true)
bookmarksopen	true/false	しおりツリーを開くか (false)
colorlinks	true/false	リンクに色を付けるか (false)
anchorcolor	色指定	アンカー文字列の色 (black)
linkcolor	色指定	内部参照リンク文字列の色 (red)
citecolor	色指定	文献番号の参照色 (green)
filecolor	色指定	ローカルファイル参照リンクの色 (magenta)
urlcolor	色指定	外部参照 URL リンクの色 (cyan)

表 4.1 パッケージ hyperref のオプション指定 (一部)

パッケージ hyperref を利用する本来の目的は、URL 表記の簡便さと美しさ、ならびにインターネット情報への[ハイパーリンク](#)を可能にすることです。文中に URL を正しく組版するには、`\url` を使って次のように書きます (間違えないように Web ブラウザなどからコピーした文字列を貼りつけます)。

```
\url{URL}
```

組版処理すると、特殊記号が含んだ URL であっても正しく、しかも**タイプライター体**でわかりやすく組版されます (表 table:hyperref のオプション urlcolor を指定すると文字列の色をデフォルトの cyan を変えることができます)。組版されて生成された PDF ファイルの閲覧時に URL 文字列をクリックすると目的のインターネット情報をアクセスできます。

Web ページのようにアンカー文字列をクリックして目的のインターネット情報にアクセスできるように組版するには、`\href` を使って次のように URL とアンカー文字列を併せて指定します (表 table:hyperref のオプション anchorcolor を指定すると文字列の色をデフォルトの black を変えることができます)。

```
\href{URL}{アンカー文字列}
```

節 4.2.1 で取り上げたように、インターネット上の書誌情報として URL 表記がパーセント符号化され得るときには、次のようにして復号化された URL をアンカー文字列としてパーセント符号化 URL にハイパーリンクすることが考えられます。

```
\bibitem{natsume_wiki} Wikipedia, 「夏目漱石」,
\href{https://ja.wikipedia.org/wiki/%E5%A4%8F%E7%9B%AE%E6%BC%B1%E7%9F%B3}
{https://ja.wikipedia.org/wiki/夏目漱石}.
```

このように組版するように努めると、印刷した場合でも目的の URL を‘読む’ことが可能になり、しかも PDF ファイルを直接閲覧している場合は直ちに目的の情報アクセスできます。

L^AT_EX で作成する固定レイアウトの文書作成において、今日では印刷物と同時に PDF ファイル閲覧の双方で読者が快適に読めるような配慮が必要です。たとえば、国連人権高等弁務官事務所 (OHCHR: Office of the United Nations High Commissioner for Human Rights) が提供している 500 以上の各言語別の「世界人権宣言」を書誌情報とするときには次のように書くとよいでしょう。

```
\bibitem{ohchr_humanright} 国連人権高等弁務官事務所,
「世界人権宣言」(1948 年 12 月 10 日),
\url{http://www.ohchr.org/EN/UDHR/Pages/SearchByLang.aspx}.
```

あるいは、日本語の人権宣言を書誌情報とするときには、さらに‘世界人権宣言’アンカー文字列として日本語ページにハイパーリンクしておきます。

```
\bibitem{ohchr_humanright}
国連人権高等弁務官事務所,
「\href{http://www.ohchr.org/EN/UDHR/Pages/Language.aspx?LangID=jpn}
{世界人権宣言}」(1948 年 12 月 10 日),
\url{http://www.ohchr.org/EN/UDHR/Pages/SearchByLang.aspx}.
```

こうした書誌情報に対する配慮は、Web ページだけでなく印刷物である書籍についても可能です。図 4.1 において、図書館に収蔵されている書誌情報へハイパーリンクしておくことで読者の利便はさらに高まります。次の例では、書籍名をアンカー文字列として国立国会図書館サーチ (<http://iss.ndl.go.jp>) から得られる詳細な書誌情報にリンクしています。

```
\bibitem{human_rights} 高木八尺, 末延三次, 宮沢俊義編,
『\href{http://iss.ndl.go.jp/books/R100000002-I000000962427-00}{人権宣言集}』,
岩波文庫, 1957.
```

4.3 参考文献の品質

4.4 相互参照を使った正確な文書

L^AT_EX では章 (chapter) や節 (section) の見出しや図表および数式などの要素番号やそれらが登場するページ数を文中の任意の場所で参照して組版することができます。この機構を相互参照と呼んでいます。

相互参照は**正確な文書** (accurate document) の作成を支援します。正確な文書とは綴り方や文法的に正当な

言語の運用を意味するものではありません。文章内で文書内の別の箇所にある文章や図表に言及する場合を考えてみましょう。‘前の文’や‘後の文’とか‘前の図’や‘後の表’という表現は言及する対象が文書のどの箇所（どのページ）にあるかは明示されておらず曖昧で、場合によっては読者に誤解を生じさせてしまうかもしれません。このような前後左右といった相対的な方向位置によらずに、文書内の絶対位置が特定できるように‘節 3.5’とか‘図 7’や‘表 2’というように書かれているときに正確な文書というのです。

既に見てきたように L^AT_EX では組版処理も手間をかけることによって、文書クラスに応じて `\chapter` または `\section` や `\subsection` などで見出し番号を付け（節 2.4）、文献番号を割り当てたり（節 4.1）、図表に番号を付ける（節 4.4.1）作業を自動化しています。正確な文書を書くためには、さらに文書の任意の場所で目的の箇所を参照して該当する番号を組版する必要があります。この作業を文書作成途中で手作業で行うことは容易ではありません。

4.4.1 ラベル指定とその参照

文書内の別の場所から参照したい文書中の箇所（通常は、参照要素の先頭）に、`\label` を使って次のように参照ラベルをキーとして定義してマークします。ラベル名は L^AT_EX 文書全体で一意的でなければならず、重複してはいけません。

```
\label{参照ラベル名}
```

この `\label` コマンドは組版出力には何の影響も及ぼしません。このラベル名によって文書内の任意で箇所から要素番号やページ数を参照します。文書中でこの参照ラベルに割り当てられた番号またはページ数を参照するには次のように書きます。

```
\ref{参照ラベル}  
または  
\pageref{参照ラベル}
```

`\ref` を使うとラベルされた要素番号を、`\pageref` を使うとラベルされたページ数が参照されます。参照ラベルを使った参照方法にはこれ以外にもバリエーションがあり、表 4.2 にまとめました。

以下で、参照ラベルをマークする場所とその参照方法を具体的に見てみましょう。参照時には番号だけが組版されるため、文章として意味が通るように参照の前後に‘章’や‘節’、‘図’や‘表’あるいは‘式’などを適宜補って記述する必要があることに注意して下さい。

参照結果を文書中に取り込むためには、続けて 2 回の組版処理が必要になります。1 回目の処理で参照番号や参照ページなどの参照情報が拡張子 `.aux` が付いたファイルに書き込まれます。次いで 2 回目の処理で `.aux` ファイルから参照情報を読み込んで文書に組版されます。

章・節番号の参照

見だしコマンド `\chapter{...}`、`\section{...}`、`\subsection{...}` など（節 2.4）を使ったとき、要素番号である章や節の見出し番号をマークするには、次のように書きます。

```
\section{さるかに合戦の背景}  
\label{background}
```

参照される対象	参照ラベルの指定	参照方法
章・節などの番号	<code>\label{ラベル}</code>	<code>\ref{ラベル}</code>
図や表	<code>\label{ラベル}</code>	<code>\ref{ラベル}</code>
数式番号	<code>\label{ラベル}</code>	<code>\eqref{ラベル}</code>
ラベルの登場ページ	<code>\label{ラベル}</code>	<code>\pageref{ラベル}</code>
文献番号	<code>\bibitem[opt]{ラベル}</code>	<code>\cite[注記]{ラベル}</code>

表 4.2 相互参照のためのラベル指定とその参照方法。参照時には番号だけが組版されるため、文章として意味が通るように参照の前には‘章’や‘節’、‘図’や‘表’あるいは‘式’などを適宜補って記述する。

```
.....
\subsection{カニの猿関係}
\label{relation}
.....
```

このように記述しておく、文書中でこれらの参照ラベルを次のように利用できます。

節`\ref{background}`では合戦の背景について考察する。
とくに、節`\ref{relation}` (`\pageref{relation}`ページ) で言及すること
であるが、カニの日ごろからの猿関係は合戦に至る主要な伏線となっている。

図表の参照

本書では図表の作成については詳しく触れませんが(奥村 [1]などを参照してください)、図表のために `figure` 環境または `table` 環境を使います。このとき、図表本体の記述以外に図表を説明するためのコマンド `\caption{...}` を使って図表の内容を説明すると、登場順に図または表番号が振られます。このとき、次のように `\caption` の後に `\label` を使ってラベルをマークするように書きます。

```
\begin{figure}[htbp]
\begin{center}
図の記述または画像の貼り込み
\end{center}
\caption{東京タワーからスカイツリーを望む}
\label{fig-photo_skytree}
\end{figure}
```

このように記述しておく、文書内でこの参照ラベルを次のように利用できます。

図`\ref{fig-photo_skytree}` (`\pageref{fig-photo_skytree}`ページ) は、
東京タワーからスカイツリーを望む写真である。

数式の参照

L^AT_EX 文書の大きな特徴の一つとして美しい数式表現が組版できることを挙げることができます (節 6)。標準の数式行環境 `equation` を使って数式番号を表示させてこれをマークするには次のように書きます。

```
\begin{equation}
\label{gauss_integral}
\int_{-\infty}^{\infty} \mathrm{e}^{-ax^2} dx = \sqrt{\frac{\pi}{a}}, \quad a > 0
\end{equation}
```

このように記述しておく、文書内でこの参照ラベルを次のように利用できます。

式`\eqref{gauss_integral}`を Gauss 積分といい、広く利用されている。

4.5 正確な文書作成はなぜ必要なのか

相互参照を使って文書を正確に書く必要性を改めて確認しておきましょう。通常、文書は以下に見るように複雑な内部構造を有しています。その構造を明らかに読者に提示し、しかも文書全体として有機的まとまりを保つために相互参照の機構を活用するのです。

図 4.2 は直線的な文書構成を持つ文書を表しています。数学的証明を表した文書のように、各文書単位の間には前後関係 (順序関係) があり、先頭から順番に読み進めると直線的に結論に自然に導かれるような文書です。言い換えれば、文書単位の順番を入れ替えてしまうと文意が損なわれてしまう、文意が正しく伝わりません。



図 4.2 直線的な依存性を持つ文書構成。各構成単位が一行に並ぶように論理によって順序付けられている文書では、数学的証明のように、その単位を入れ替えてしまうと文意不明になってしまう。このような論述形式を持つ文書は実際には多くはない。

実際には図 4.2 のような直線的な構成となる文書は多くはなく、文書単位が図 4.3 のような木構造をなすように配置されているのが一般的です。木構造の各頂点の下に配置される文単位の間には必ずしも順序関係はなく、併記されていることがあります。図 4.3 は、主題について政治・経済および環境の側面から関連する項目を取り上げて文書が構成されている様子を表しています。着目する論点によって関連項目はさまざまにあり得て、それらの項目間に常に必然的な関係性があるとは限りません。

図 4.3 のような木構造を文書配置においては、各文書単位は階層を越えて互いに関連し、文書単位全体は複雑なネットワークをなしています。図 4.4 は、主題を政治・経済および環境の 3 つの側面から検討を進めた際に、それらの項目要素が互いに深く結びついて関係し合っている様子を示しています。このような文書の論述においては、相互参照によってその内部構造を明確にすると同時に、文書全体として有機的まとまりが保持されるように細やかな配慮が必要となるのです。

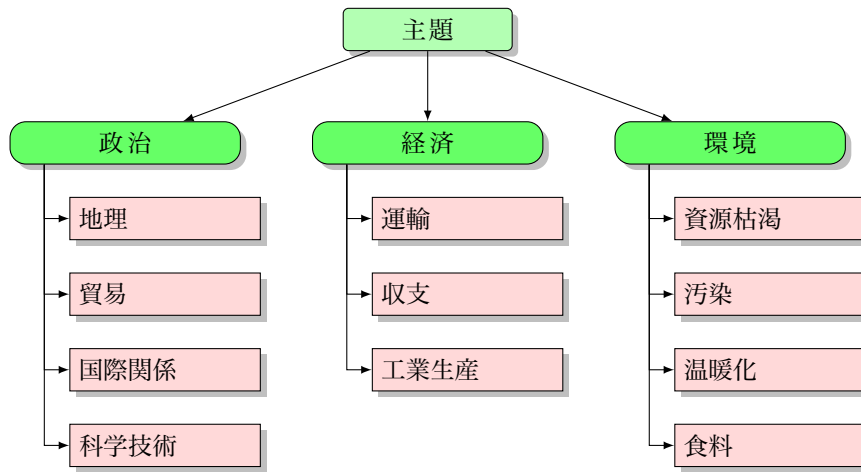


図 4.3 文書の主題を政治、経済、環境の各側面から考えた文書構造の例。文書単位が木構造をなるように配置され、文単位は完全な前後関係をもたない。

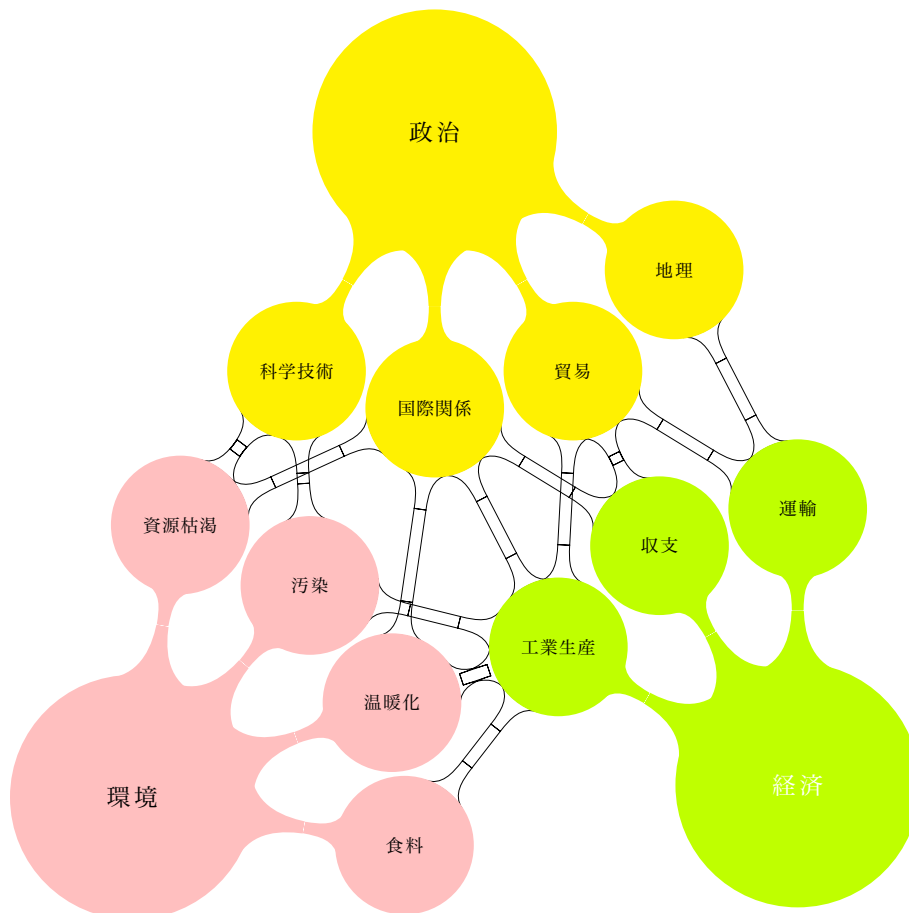


図 4.4 政治、経済、環境における相互依存性。図 4.3 のような木構造をなして完全な前後関係を持たない文書構成では、文単位全体は複雑な相互依存のネットワークをなしている。考察対象の取り上げ方に応じて関連項目とその依存性はダイナミックに変化する。

第 5 章

作表と画像の取り込み

5.1 簡単な作表

作表の基本は `tabular` 環境を使います。例えば、

1	2	3
左側	中央部	右側
Apple	Pine	Orange

 のように表を作成することができます。文中の表部分は次のように書かれています。

```
\begin{tabular}{lcr}
\hline
1 & 2 & 3\\
\hline
左側 & 中央部 & 右側\\
Apple & Pine & Orange\\
\hline
\end{tabular}
```

5.1.1 図表の出力位置

`tabular` 環境を使ったままでは、本文中に表が大きな 1 文字のように扱われてしまうので、通常は次のように `table` 環境内に置きます。このとき、`tabular` 環境をさらに `center` 環境ではさむと中央に位置させることができます。

1	2	3
左側	中央部	右側
Apple	Pine	Orange

表 5.1 簡単な表

連番が付く図表のために、 \LaTeX では図の場合には `figure` 環境、表の場合には `table` 環境が用意されており、以下の書式に従います。

```

\begin{table/figure}[出力位置指定]
画像の張り込みや作図または tabular 環境などによる作表
図表を中心位置に配置するには center 環境で挟む
\caption{図表の説明文}
\label{ラベル名}
\end{figure/table}

```

figure/table 環境は float 環境と呼ばれる仲間に属し、その出力位置は状況によって半自動的に決定され、ページ内または後のページへと動きます（それゆえ float なのです）。表 5.2 に figure/table 環境で使われる図表の‘出現位置指定’で使われるパラメータ文字の意味を示しました。

位置指定	文字の意味
h	できるだけ tabular の出現場所へ出力する
t	できるだけページの先頭へ出力する
b	できるだけページの末尾へ出力する
p	図表だけからなるページを作成して、そこに出力する

表 5.2 figure/table 環境で使われる出力位置指定パラメータとその意味

この位置指定は、float な図表の出力位置に対する努力目標としての意味しかないことに注意してください。たとえば、[htb] と指定すると、できれば‘この’位置に、次いでページトップに、それでも無理ならページボトムにという意味になります。したがって、TeX では \label{ラベル名} でラベル名を指定し、\ref{ラベル名} でそのラベルを参照して図表の番号を取得しながら文を書くようにするのがよいのです。

図表の位置をここだと‘絶対指定’したければ、プレアンブル部で

```
\usepackage{here}
```

とパッケージ here を読み込んだ上で、figure/table の出力位置指定で [H] とすれば、「その」場所へ出力されるようになります。

5.1.2 tabular 環境の書式

tabular 環境は次のような書式を持ちます。

tabular 環境

```

\begin{tabular}{位置書式}
.....
表項目の並び
.....
\end{tabular}

```

ここで、位置書式（左右寄せパラメータ）には次のような指定が可能です。表の各行の項目数は位置書式で指

定した数以下でなければなりません。項目間は記号 & で区切り、各行の終わりには2つのバックスラッシュ (強制改行コマンド) \\ を書きます。強制改行までの項目数が不足していれば、残りは空白項目として扱われます。

tabular 環境内の項目として表 5.4 のものが利用できます。

5.1.3 作表における技巧

tabular の列の位置書式 (左右寄せパラメータ) を指定する引数内で p{ 幅指定 } を使って幅指定した段落モードにすることができます。段落の幅は、たとえば 0.3\textwidth とするとテキスト幅の 30% となります。

記号	意味	備考
l	項目を左寄せ (Left) にする	小文字の L
c	項目を中寄せ (center) にする	
r	項目を右寄せ (Right) にする	
	項目間に縦罫線を引く	縦棒
	項目間に 2 重縦罫線を引く	縦棒 2 本

表 5.3 tabular 環境の位置書式

<code>\hline</code>	行の先頭 (または最後) だけに書くことができる。 <code>\hline</code> が置かれている「場所」から、表を横切る横罫線を表の幅の最後まで引く。 <code>\hline\hline</code> と続けると 2 重横罫線になる。
<code>\vline</code>	縦罫線
<code>\cline{start-end}</code>	start 番目から end 番目までのカラムに横罫線を引く。
<code>\multicolumn{num}{pos}{item}</code>	複数のカラムにまたがる項目を作成するときを使う。num はまたがるカラム数で、それで確保された空間に pos で指定された水平方向の位置に、項目 item を置く。pos は l (L 小文字), c, r の何れかの 1 文字を含み、 (縦棒) を含んでも構わない。

表 5.4 tabular 環境における特別な表要素

雨の日の注意事項	傘と雨具、レインシューズをはくのはもちろんですが、雨の日は足元が滑りやすくなり危険なことが多くなります。
晴天の日の注意事項	十分な水分を取り、着替を持参する。ただし、水分を過剰に摂りすぎたり、喉越しはよいけれど栄養価の低いものばかり食べてしまったり、体に負担をかける飲食には注意します。

表 5.5 段落モードを利用した表 (2 列目が段落モード。幅は横幅の 40% とした)

表 5.1 では、1 つの表だけを置いたが、表が小さければ次のようにすることも可能です。tabular の列の左右寄せパラメータを指定する引数内で p{} を使って幅指定した段落モードにした上で (真ん中の段落は&&で空の列要素としている)、それぞれの列要素として tabular 環境で作成した 1 行の表 (tabular の並び要素が tabular) としています。それぞれの段落内で\centering を使って中央揃えとしています。

1	2	3
左側	中央部	右側
Apple	Pine	Orange

表 5.6 左側の表

1	2	3
左側	中央部	右側
Apple	Pine	Orange

表 5.7 右側の表

最後に、\multicolumn と\cline を使った例を挙げておきましょう。

シュークリームの材料		
品名	分量	用途
バター	100g	シュー生地
塩	2g	
小麦粉	100g	
卵	3 個	
卵黄身	4 個分	カスタードクリーム
砂糖	100g	
小麦粉	50g	
ミルク	500cc	
バニラエッセンス	少々	
洋酒	少々	

表 5.8 シュークリームのレシピ

5.2 画像ファイルの取り込み

文書中にテキストや図の拡大縮小・回転を行うためのパッケージ `graphicx` を使って、別に用意した画像ファイルを貼り込んで組版することができます。詳しい情報は、たとえば TeXWiki の [TeX 入門/図表](#) にあります。

TeXWorks や TeXShop などの L^AT_EX 統合環境の一般的な設定を使って組版する場合には詳しく知っておく必要はありませんが、正確に言うと画像ファイルの取り込みは L^AT_EX 固有の機能ではありません。画像を貼り込む組版では、まず L^AT_EX が別のプログラム処理によって得られた **バウンディングボックス (BoundingBox)** 情報という画像サイズに関するパラメータを取得して画像を貼り込む組版領域を確保した上で、**ドライバ**と呼ばれるプログラムが画像ファイルを取り込んで目的とする PostScript または PDF 形式の組版ファイルを生成します。

PDF ファイルは Adobe 社によって開発された文章・画像の閲覧ファイル形式ですが、国際標準化機構 (ISO) によって標準化され、特定の環境に左右されずに全ての環境でほぼ同様の状態で閲覧できます。本書でも L^AT_EX 組版処理によって最終的に生成されるファイルを PDF としており、L^AT_EX 統合環境を利用して L^AT_EX 文書を組版する場合にドライバ `dvipdfmx` を使うやり方が広く普及しています。

画像ファイルを取り込んで PDF ファイルとして組版する場合には、プリアンブルでパッケージ `graphicx` の利用宣言の際に次のようにドライバ `dvipdfmx` を使うオプションを指定します。節 4.2.2 で PDF ファイルにリンク情報を埋め込むためのパッケージ `hyperref` を紹介しましたが、このときも同じく `hyperref` の利用宣言の際にオプションにドライバ `dvipdfmx` を指定しました (35 ページ)。

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

パッケージ `graphicx` の利用宣言をした後に、本文中の画像を貼り込みたい箇所で画像を貼り込んで組版するには次のように `\includegraphics` を使って目的の画像ファイルを指定します。

```
\includegraphics[オプション]{画像ファイル名}
```

ファイル名の指定では、組版処理の対象である L^AT_EX のルートファイル (節 2.6 参照) から目的の画像ファイルへのパスを含めます。フォルダごとのファイルパスの区切り文字はスラッシュ `/` を使います。作成する文書ごとに L^AT_EX ファイルを格納するフォルダを作成し、その中にさらに画像ファイル専用のフォルダを用意しておくことをお勧めします。

画像形式	略記名	拡張子
Portable Network Graphics	PNG	.png
Joint Photographic Experts Group	JPEG	.jpg .jpeg
Portable Document Format	PDF	.pdf
Encapsulated PostScript	EPS	.eps .epsf

表 5.9 パッケージ `graphicx` で読み込み可能な画像形式

`graphicx` を使って貼り込める画像ファイルの形式を表 5.9 に示しました。PNG は GIF 形式に代わって開発された可逆圧縮形式、JPEG は主に写真画像を対象とする非可逆圧縮形式です。EPS は PostScript で記述され

た図版データを画像ファイルとして保存するためのファイル形式です。表 5.9 以外に広く利用されている画像形式、たとえば GIF(Graphics Interchange Format: 拡張子 .gif) や SVG(Scalable Vector Graphics: 拡張子 .svg) など、また Windows でしばしば使われている BMP(Bitmap Image: 拡張子 .bmp) 形式の画像ファイルは、各種の画像ソフトウェアまたは画像変換 Web サービスを使ってパッケージ graphicx で取り扱える画像形式に変換しておきます。画像を取り扱う場合にはそのファイル拡張子に注意を払い、その画像形式を知っておくようにしてください。

5.2.1 画像貼り込みオプション

L^AT_EX とドライバ dvi_pdfmx を使ってパッケージ graphicx を利用して画像ファイルを貼り込むためには、まず画像サイズに関する BoundingBox 情報のパラメータ (画像が配置されている 4 隅のピクセル位置) を取得する必要があります。画像ファイル閲覧ソフトウェアで確認することもできますが、最近の L^AT_EX システムでは、組版処理の途中で外部コマンド extractbb が自動実行されるように設定されているため、事実上は BoundingBox 情報の取得を気にかける必要はありません。図 5.1 は、extractbb を画像ファイル my_picture.png に対して実行して得られる xbb ファイルの内容で、3 行目に BoundingBox 情報が書き出されています。この画像は 521 × 619 ピクセルの大きさを持つことがわかります。

```
%%Title: my_picture.png
%%Creator: extractbb 20160307
%%BoundingBox: 0 0 521 619
%%HiResBoundingBox: 0.000000 0.000000 520.934883 618.922635
%%CreationDate: Sun Dec 4 18:10:54 2016
```

図 5.1 外部コマンド extractbb を使って画像ファイル my_picture.png から作成した xbb ファイルの内容

画像ファイルを読み込むコマンド `\includegraphics` のオプションの一部を は表 5.10 に掲げました。オプション指定をしなければ、読み込んだ画像そのままのピクセル数の大きさで張り込まれます。作成する文書の大きさや目的に応じて調整してください。複数のオプション指定はカンマ ‘,’ で区切って並べますが、カンマの前後には余分な空白を入れてはいけません。

ルートドキュメントが位置する場所にフォルダ image を作成した画像ファイル my_picture.png を納めている場合、これを読み込む例を示してみます。

```
\includegraphics[height=4cm,width=3cm]{image/my_picture.png}
```

この例では、2 つのオプション height と width を使って、読み込んだ画像を高さ 4cm 幅 3cm で組版します。この場合、元の画像の縦横比は変わってしまっています。次の例は、読み込んだ元の画像が高さ 4cm になるように縮尺を調節して組版します。

```
\includegraphics[height=4cm]{image/my_picture.png}
```

元の画像の縦横比率を保って拡大率を変えて 0.7 倍に縮小して組版するには次のように書きます (scale=1 で等倍になります)。

```
\includegraphics[scale=0.7]{image/my_picture.png}
```

オプション	意味
height	組版する画像の高さ指定
width	組版する画像の幅指定
scale	組版する画像の縮尺指定
angle	組版する画像を回転 (単位は度)

表 5.10 パッケージ `graphicx` の画像読み込みコマンド `\includegraphics` のオプション (一部)

5.2.2 図を相互参照する

`\includegraphics{...}` で張り込まれる画像は ‘文中の 1 文字’ として組版されます。貼り込んだ画像を文書の図として取り扱う場合には **figure 環境** を使って次の要領で記述します。

```
\begin{figure}{h}  
\begin{center}  
\includegraphics[オプション]{画像ファイル名}  
\end{center}  
\caption{図の説明}  
\label{参照ラベル}  
\end{figure}
```

`center` 環境は読み込んだ画像を行方向のページ中央に配置するために使っています。図番号を付けて組版するために `\caption` で図の説明文を書きます。4.4.1 節で説明した図番号を相互参照するために、`figure` 環境内の最後の行に `\label` によって参照ラベル名を指定してマークします。任意の文章中から `\ref{ラベル名}` によって図番号を、`\pageref{ラベル名}` によってマークしたページ数を参照することができます。

第 6 章

数式を書く

6.1 2つの数式モード

6.2 簡単な数式

6.3 数式を組み立てる

第 7 章

索引の作成

L^AT_EX では文書内に索引情報を埋め込んでおき、索引項目をその読み方で並べ替えて登場ページ数とを一覧表示する**索引作成**を自動化することもできます。

MakeIndex という索引作成のソフトウェアを利用して索引の作成を自動的に行なうことができます。索引の作成は通常の本編編集において最も手間のかかる作業の一つです。ここでは MakeIndex を ASCII が日本語化した mendex を使った索引の作り方を説明します。

7.0.1 索引作成の手順

索引を作成するには L^AT_EX ファイルには次の記載が必要です。

- ▷ プリアンブルにパッケージ `makeidx` を読み込む
- ▷ 続けてプリアンブルに `\makeindex` を宣言
- ▷ 本文中で索引項目を `\index{よみ@読み}` によって指定
- ▷ 文中で索引を出力する位置にコマンド `\printindex` を記入

具体的には次の形式のファイルを作成します。

```
\documentclass{文書クラス}
\usepackage{makeidx}
\makeindex
....
\begin{document}
....
\index{さくいんこうもく@索引項目}
....
\printindex
\end{document}
```

本文中で索引項目を `\index{..@..}` によって指定してあっても、文書の出力には何の影響もありません。したがって、将来索引を必要とする可能性がある場合は当然として、文書処理においては `\index{..@..}`

を検索時のキーワードとして利用できるという利点もあるために、できる限り`\index{..@..}`を使って索引項目を選び出して書いておくとよいでしょう。

索引の自動作成を達成するためには、次の手順に従って必要な回数のコンパイル作業が必要です。

1. ファイルをコンパイル（目次の挿入するときには、ページ数がずれるので最低2回コンパイル）し、拡張子`.idx`の付いた索引情報ファイルを作成します。
2. 次に日本語 MakeIndex ソフトウェアである `mindex` を使って `idx` ファイルを処理して項目がアルファベット順と 50 音順に並べてページ番号に対応させた拡張子`.ind`の付いた**索引ファイル**を作成します。文書ファイルが`latexfile.tex`の場合には、`latexfile.idx`ファイルがあることを確認してから、次のように`makeindex`または`mindex`コマンドを実行します（どちらのコマンドを使うか、または両方とも使えるかは利用している TeX システムに依存します）。

```
% mindex(or makeindex) latexfile.idx
```

索引ファイル`latexfile.ind`が作成されたことを確認します。

3. もう一度、コンパイルします。索引ファイルの`ind`ファイルを読み込んで、`\printindex`が記入された位置に索引項目とその出現ページが順に印刷されます。
4. さらにもう一度コンパイルすると目次に索引ページが載ります。目次をいれて、目次に索引ページを入れるためには最低でも4回のコンパイル作業が必要です。

7.0.2 索引項目の指定

索引項目として指定するには文中で次の`\index`コマンドを使います。

```
\index{索引項目の読み方@索引項目}
```

ただし、索引項目がカタカナや漢字を含む場合には上のよう書きませんが、半角アルファベットやひらがなだけの索引項目は`index{索引項目}`とだけ書きます。

索引では、最初に英文字で始まる項目が“アルファベット順”に並び、次いで和文字で始まる項目が“50音順”に並びます。したがって、索引作成のポイントはコマンド`index{..@..}`の使い方、とくにその索引項目の読み方の指定にあります。

英数字記号 半角英数字‘だけ’が索引項目のときは上で注意したように、索引項目をそのままを指定します。記号などが混じる場合、たとえば`\LaTeX`という索引項目の読み方は‘`LaTeX`’でも‘`latex`’でも構いません。読み方については大小文字の区別はありません。

和文 カタカナや漢字が索引項目ときには、文章の‘揺れ’に注意します。たとえば、“コンピュータ”と“コンピューター”とは違う索引項目になります。また、同じ索引項目に異なる読みを付け場合には、索引の取り扱いとは別になります。

7.0.3 索引作成の文書例

索引を作成する L^AT_EX ファイルの例を次に示します。

```
\documentclass{jsarticle}% jsarticle の利用を推奨
\usepackage{makeidx}
\makeindex
....
\begin{document}
\maketitle
\tableofcontents
.....
.... さるかに合戦
\index{さるかに合戦@さるかに合戦}
に関する多種多彩な側面を深く検討することによって、従来の民話
\index{みんわ@民話}
的歴史認識
\index{れきし@歴史認識}
から得られないあらゆる研究方法の獲得を説明することができる。
例えば、さるが盗んだとされている餅
\index{もち@餅}
をめぐる考察から当時の農耕社会
\index{のうこうしゃかい@農耕社会}
システムが把握されるのである。
社会問題
\index{しゃかいもんだい@社会問題}
との関連性を説明する餅の領域に‘合戦’の影響が見られるという認識は
.....

\printindex
\end{document}
```

この例では、あえて索引項目を指定する `index{...@...}` を行頭に置いています。索引項目の確認や将来の文書処理の容易さのために、文書の実際の出力には無関係なこれらの記載はできるだけわかりやすく記述しておくためです。この文書ファイルに対して、以上の索引作成の手続きを経ると文書ファイルの最後に索引ページが出力されます。

第III部

さらなる応用

第 8 章

パッケージの利用

3.1 節でわずかに取り上げた L^AT_EX で標準に定義されている文書クラスで利用できる環境だけでは、きめ細かい文書レイアウトを行うことが難しい場合があります。このような場合には、別に用意されたマクロパッケージを次のようにプリアンブル部に `\usepackage{.}` コマンドを使って読み込みます。パッケージに関する詳しい情報は T_EXWiki を参照してください。画像ファイルの張り込みのために利用した `graphicx` パッケージについてはすでに ?? 節 (?? ページ) で説明しました。

```
\documentclass[クラスオプション]{文書クラス}
\usepackage{パッケージ名}
....
\usepackage{パッケージ名}
....
\begin{document}
....
```

8.1 段組文書

段組文書とするには、2つの方法があります。1つ目は、文書スタイルでオプション `twocolumn` を指定して 2 段組とする方法です。

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

この場合には文章全体が 2 段組で出力されます。

パッケージ `multicol` を使うと、文中の任意の場所を **多段組**文書とすることができます。たとえば、指定した範囲にある文書を 2 段組にするには、`multicol` パッケージを読み込んで次のように書きます。`multicols` 環境に渡すパラメータが多段数を指定します。

```

\documentclass{jsarticle}
\usepackage{multicol}
...
\begin{document}
.....
文章を書く。ここに書かれた文章は 1 段組で出力される。
.....
\begin{multicols}{2} % 2 段組を指定
.....
    ここに文章を書くと、2 段組で出力される。
    段数は \begin{multicol}{カラム数} で指定する
.....
\end{multicols}
ここ以降に書かれた文章は 1 段組で出力される。
.....
\end{document}

```

多段組文書は、乱用すると印刷バランスが崩れてしまい読みにくくなってしまいます。日本語では、本文のフォントサイズを小さくしない限り多段組は 2 か 3 段が見た目にも読みやすさでも限界でしょう。段組の必要性と効果をよく考えてから利用してください。

8.2 パッケージの入手とインストール

パッケージ利用は L^AT_EX の世界を大きく広げます。標準的な T_EX システムでは既に主なパッケージファイル群がインストールされています。T_EX システムがコンピュータのどの場所にインストールされるかは T_EX インストーラに依存します。Windows の「T_EX インストーラ 3」では C:\w32tex/share/以下 (津田では C:\usr/local/share/以下) にあるフォルダ texmf/内の tex または ptex 以下にインストールされています。『L^AT_EX コンパニオン』[?] には、代表的なパッケージの利用法が詳しく紹介されています。「L^AT_EX でこんなことができたなら」と思った場合に参考になるでしょう。日本語 T_EX Wiki でもさまざまなマクロパッケージのインストール法や使い方が集められていますので参考にしてください。

パッケージを含む T_EX 関係のソフトは CTAN(Comprehensive T_EX Archive Network) と呼ばれるサイト群に集められています*1。パッケージ名が分かっている場合には ‘tex_パッケージ名’ などで検索してパッケージをダウンロードします。

パッケージを定義している本体のファイルは**スタイルファイル**とも呼ばれ、拡張子 ‘.sty’ または ‘.cls’

*1 入手したいパッケージは

T_EX User Group <http://www.tug.org/>
 から、または日本のミラーサイトとして
 RING サーバ <http://www.ring.gr.jp/pub/text/CTAN/>
 から入手することができます。

が付きます。スタイルファイルはパッケージを必要とする TeX ファイルと同じフォルダに置けばよいのですが、それでは不便なので、以下で説明するしかるべき場所に配置 (インストール) し、コマンド `mktexlsr` でその場所を TeX システムに記録しておく、任意の場所にある TeX ファイルからパッケージを呼び込むことができます。

ダウンロードする際には、スタイルファイル (`.sty`, `.cls`) だけが配布される場合や関連ファイル一式として配布される場合があります。それらの関係は次のようになっています：

`.sty`, `.cls` パッケージの本体ファイル。

`.dtx` パッケージ本体と説明文書をパックしたファイルで、`ins` ファイルが必要。

`.ins` `dtx` ファイルから、スタイルファイル本体や説明文書をアンパックするためのファイル。`dtx` ファイルとセットでダウンロードする。

パッケージとしてパックされたファイルたとえば、`package.dtx` の場合、一緒にダウンロードした `package.ins` が同じフォルダにあることを確認したうえで

```
% platex package.ins
```

を実行して、アンパックします。これによって、(場合によっては複数の) `sty/cls` ファイルや説明文などが生成されたことを確認します。説明文書が含まれている場合には `dtx` ファイルから

```
% platex package.dtx
```

によって `dvi` ファイルが生成されます (何度か `platex` を適用する必要があるかもしれません)。

こうして得られたパッケージファイル群を TeX システムにインストールするには、まず TeX システムがその場所を探索可能な場所に置きます。通常はそのフォルダごと、たとえば `texmf/tex/misc` や `texmf/ptex/misc` に移動します (`texmf/tex` 内に自由にフォルダを作成して構いません)。そして、これが重要なことですが、その後にかならずコマンド

```
% texhash
```

を実行します (以前は `mktexlsr` でしたが最近では `texhash`)。これによって、TeX システムが探索可能なパッケージファイル群などを記述したファイル `ls-R` が生成されます。こうすることによって、TeX 文書がどこにあってもプリアンブル部に `\usepackage{..}` と書いてパッケージを読み込む際に、同じフォルダにスタイルファイル `sty/cls` を置かなくて済むのです (8.4 節参照)。

8.3 PDF ファイルのハイパーリンク

8.4 TEXMFHOME の利用

TeX システムのインストール時には、環境変数 `TEXMFHOME` が設定されます。`TEXMFHOME` とは、TeX ユーザ個人が自由に使うことのできるフォルダ `texmf` を置くことができる場所で、定められたフォルダ構成にしたがって、TeX システムが有する標準的なパッケージ以外のパッケージを CTAN(The Comprehensive TEX

Archive Network)^{*2} などから入手したパッケージを置いておくことができます。

もし TEXMFHOME を利用しないとすると、TeX の本文ファイルに必要な標準以外のパッケージ（つまり、スタイルファイル）をその TeX ファイルと同じフォルダに置いておく必要があります。新たな TeX ファイルを作成するために必要なパッケージファイルを同じフォルダに置くことになって（同じパッケージファイルを重複して配置することになるでしょう）、ファイル管理上、無用の混乱を来してしまいます。TeX ファイルに必要な標準以外パッケージは、以下で説明する方法にしたがってフォルダ texmf 内のサブフォルダ tex 以下（つまり、texmf/tex 以下）に配置しましょう（必要なパッケージが現在の TeX システムに存在しているのかどうかを調べる方法も以下で説明します）。

TeX システムでは、TeX システムにとってたいへん重要なさまざまな情報、特に TeX ユーザにとって必要なさまざまなパッケージ（スタイルファイル）が取められている場所が複数あり、それらは **TEXMF ツリー** と呼ばれています。これらのコンピュータ内の場所は、TEXMF で始まる環境変数で指定されています。

8.4.1 TEXMFHOME の場所

TEXMFHOME の場所は、コンピュータの OS や TeX システムの配布形態によって異なります。大抵の TeX インストーラでは、ユーザのホーム領域が TEXMFHOME の場所に割り当てられていて^{*3}、そこに次に従ってフォルダ tt texmf を作成します。

Windows

- Windows XP の場合: C:\Documents and Settings**ユーザー名**
- Windows Vista の場合: C:\Users**ユーザー名**
- Windows 7/8 の場合: C:\Users**ユーザー名**

Mac

- Macport で TeX システムを構築した場合、TEXMFHOME は ~/（ユーザホーム）に設定される。
- MacTeX によって TeX システムをインストールした場合には、TEXMFHOME は ~/Library に設定される。ただし、MacOS 7(Lion) からフォルダ ~/Library は不可視になっています。MacOS 10.8 Mountain Lion のユーザライブラリを可視化する方法などを参考に見えるようにしてから、Desktop でフォルダ TEXMFHOME/texmf を作成してから、~/Library にドラッグするようにすると間違えないでしょう。

8.4.2 TEXMFHOME の使い方

上で説明したそれぞれの TeX システムに応じた TEXMFHOME に texmf という名前のフォルダを作成します（これを以降で TEXMFHOME/texmf と表記します）。ただし、フォルダ TEXMFHOME/texmf は次のような構造をになっていなければなりません。texmf 内にサブフォルダ tex を、そのサブフォルダ tex フォルダがなければならず、必要ならその中に platex や latex など任意のフォルダを置くことができます）。

TEXMFHOME（←インストール環境に応じて場所が決まっています）

```
texmf
  |--tex
```

^{*2} <http://www.ctan.org>

^{*3} TeX インストーラ 3 <http://www.math.sci.hokudai.ac.jp/~abenori/soft/abtexinst.html> もそうです。

```

|-- 直接パッケージを置いてもよい
  |--platex <-- 日本語に関わるパッケージ (フォルダごとでもよい)
  |--latex  <-- 一般のパッケージ (フォルダごとでもよい)
  |--misc   <-- 何か他のパッケージ

```

TEXMFHOME/texmf/の中には標準以外の TeX パッケージ (スタイルファイル) を置きます (必要なフォルダごと置いて構いません)。

すると、TeX システムは TEXMFHOME/texmf/tex/ を含む TEXMF ツリー内を検索して、それが存在すれば、TeX ファイルで `\usepackage{...}` で宣言したパッケージとして読み込んでくれます。とても便利!

大学のシステムのように、ユーザがホーム以外に自由にパッケージを追加できない場合、TEXHOME に texmf フォルダを作成しておくことはとても大切です。そのとき、追加するパッケージは TEXHOME/texmf/tex に置かねばならないことを再度強調しておきます (TEXMFHOME/texmf/ 内にパッケージを置くと探すことができません)。

8.4.3 TeX システム内のパッケージを探す

必要なパッケージ名がわかっているときに、現在の TeX システムにそのパッケージが存在しているかどうかを知るには、コマンド `kpsewhich` を次のように使います (下の記号「%」はコマンドプロンプトであり、入力する必要はありません)。もしパッケージが存在しない場合には、インターネット経由でパッケージを入手して、自分の TEXHOME/texmf/tex/ 内に置きます。次の例は、MacTeX の場合に、パッケージ (スタイルファイル) `fourier.sty` や `pxjahyper.sty` が TeX システムの TEXMF ツリー内にあるかを調べた例です。

```
% kpsewhich fourier.sty<
/usr/local/texlive/2012/texmf-dist/tex/latex/fourier/fourier.sty
```

```
% kpsewhich pxjahyper.sty
/Users/masahiro/Library/texmf/tex/misc/PXjahyper-master/pxjahyper.sty
```

この例では、それらのスタイルファイルは存在して、そのファイルの場所はそれぞれ `/usr/local/texlive/2012/texmf-dist/tex/latex/fourier/fourier.sty` および `/Users/masahiro/Library/texmf/tex/misc/PXjahyper-master/pxjahyper.sty` にあると表示されました。

TeX システムがどの TEXMF ツリーを検索するかを確認するには、この `kpsewhich` コマンドを次のように使います。

```
% kpsewhich -var-value TEXMF
{C:/Documents and Settings/masahiro/texmf</font>,C:/w32tex/share/texmf-projects,
C:/w32tex/share/texmf-local,C:/w32tex/share/texmf} <- Windows XP の場合
```

```
{/Users/masahiro/Library/texlive/2012/texmf-config,/Users/masahiro/Library/texlive/2012/
/Users/masahiro/Library/texmf,!!/usr/local/texlive/2012/texmf-config,
```

```
!!/usr/local/texlive/2012/texmf-var,!!/usr/local/texlive/2012/texmf,  
!!/usr/local/texlive/2012/./texmf-local,  
!!/usr/local/texlive/2012/texmf-dist} <- MacTeX の場合
```

自分で作成した TEXMFHOME/texmf が検索対象のフォルダになっていることをまず確かめてください。繰り返しますが、パッケージは上で説明したように TEXMFHOME/texmf/tex/ 等の中に置いておかねばなりません (パッケージを含むフォルダ全部でもよい)。

8.5 描画パッケージ TikZ

第9章

縦組文書

9.1 縦書き文書クラス

日本語の縦組用の文書スタイルには `tarticle` と `tbook` の2つがある。次は縦組論文の場合で、通常の論文スタイルと `\documentclass{tarticle}` の一行だけが異なっている。

```
¥ documentclass{tarticle}
... プリアンブル部
¥ title{題名}
¥ author{著者名}
¥ date{日付}
¥ begin{document}
¥ maketitle

... 本文
.....
\end{document}
```

9.2 ルビをふる

漢字などにルビをするためのパッケージには `ruby.sty`^{*1} または `furikana.sty`^{*2} などがある。

```
----- ruby.sty -----
\usepackage{ruby} % プリアンブル部で宣言

\ruby{熟語}{ふりがな}
```

```
----- furikana.sty -----
\usepackage{furikana} % プリアンブル部で宣言
```

^{*1} `ruby.sty` <http://www.nls.ics.saitama-u.ac.jp/~tohru/ja/Exports/External/Chosho/ruby.sty>.

^{*2} `furikana.sty` <http://homepage3.nifty.com/xyntex/fujitas2/texlatex/tategumi/furikana.sty>.

```
\kana{熟語}{ふりがな}
```

9.3 脚注

横組文書における脚注`\footnote{...}`は、その‘列’の下でなく、縦組では左端または最後ページに追い込まれて具合が悪い。この不具合を修正するパッケージに `kyakuchu.sty`^{*3} があり、次の書式に従う。まず、マーク付きで脚注本文を `kyakuchutext[マーク]{脚注文}` で定義しておき、それ以降の本文の箇所では `\kyakuchumark{マーク}` によってマークを参照して脚注を付けるのである。

```
kyakuchu.sty
\usepackage{kyakuchu} % プリアンブル部で宣言

\kyakuchutext{脚注マーク 1}{脚注文}
\kyakuchutext{脚注マーク 2}{脚注文}
...
\kyakuchumark{脚注マーク 1}脚注を付けたい本文
...
文献\kyakuchumark{脚注マーク 2}で解説....
```

パッケージ `kyakuchu` を使った具体例を以下にします。`\footnote{...}` も使っているので、その効果を検討されたい。

```
¥ documentclass[a4j]{article}
¥ usepackage{furikana}
¥ usepackage{kyakuchu}
¥ begin{document}
歌枕
\footnote{
歌枕とは、和歌に引証される地名のこと。
}として、
\kyakuchutext{A1}{福島県白河市にあった奥州街道の関所。}
\kyakuchutext{A2}{芭蕉「おくのほそ道」
萩原 恭男 校注、岩波文庫七九（一九九一）。}
\kyakuchutext{A3}{蓑笠庵 梨一「奥細道菅菰抄」（おくのほそみちすがもしょう。
文献\ref{A2}に付録として掲載）の注釈が、
典拠を明らかにしている。}
\kyakuchumark{A1}白河の関は古来有名である。
ここより外は\kana{陸奥}{みちのく}として、
人々の旅情をかきたてる場所であった。
```

^{*3} `kyakuchu.sty` <http://homepage3.nifty.com/xymtex/fujitas2/texlatex/tategumi/kyakuchu.sty>.

松尾芭蕉\kyakuchumark{A2}「奥の細道」
の白河（白川）の関の条には、この歌枕を読み込んだ
\kyakuchumark{A3}古歌の一節がさりげなく引用されている。
\end{document}

第 10 章

スライドおよびポスターの作成

TeX システムを使ってプレゼンテーションのためのスライドを作成する数多くのパッケージが提案されてきました。TeX システムで作成するスライドは今日では PDF ファイルとして作成し、Adobe Reader でフルスクリーンとして表示してプロジェクトを通してプレゼンテーションをおこないます。PDF ファイルとしてスライド（やそのハンドアウト）を作成すると、特定の OS や有料ソフトウェアに依存せずに配布ができるという利点があります。

ここではパッケージ Beamer^{*1} をつけたスライドとポスターの作成を簡単に紹介します。

10.1 Beamer でスライド

Beamer パッケージを使ったスライドは、現在 L^AT_EX で作成するスライドの事実上の標準となっており、多くのデザインテーマの提供（それらを改造して自分専用のテーマ作成も可能です）やページリンク機能およびアニメーションなど、専用のプレゼンテーションソフトウェアに匹敵するインタラクティブなスライドを作成することもできるようになっています。

以下のソースは platex でコンパイルし、DVI ファイルを dvipdfmx によって PDF ファイルを生成してスライドを作成する Beamer ソース例です（スペース節約のために 2 段組としました）。

一行目の `\documentclass[dvipdfm]{beamer}` で beamer パッケージを読み込んでいますが、オプションで `dvipdfm` を指定していることに注意して下さい。英語のみの L^AT_EX 文書の場合、海外では `pdflatex` を使って直接 PDF ファイルを生成するのが標準的なのですが、`pdflatex` はまだ日本語に対応していません。したがって、日本語 L^AT_EX 文書では `platex + dvipdfmx` を使わざるを得ず、beamer ではこのオプション `[dvipdfm]` を指定する必要があります^{*2}。

また、2,3 行目で `\usepackage{hyperref,pxjahyper}` でパッケージ `hyperref` と `pxjahyper` を読み込んでいます。これで、`hyperref + dvipdfmx` の組み合わせで日本語を含む「しおり」をもつリンクが埋め込まれた PDF スライドを作成することができます。Beamer テーマとして、この例では Madrid を使っていますが、多くのテーマが標準で用意されています。いろいろ試みて下さい。

Beamer で作成するスライドでは、スライド 1 枚分を次のように `frame` 環境内で記述します。

```
\begin{frame}{スライドタイトル}
```

^{*1} The beamer package <http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/>.

^{*2} L^AT_EX サンプルが上手くコンパイルできない場合にも、`documentclass` でこの `[dvipdfm]` オプションを指定してみてください。

```

...
ここに 1 枚分を書く
...
\verb+\end{frame}

```

ここで frame 環境が続いて {スライドタイトル} は、各スライドの上部にスライドタイトルを表示するためです ({スライドタイトル} 自体を省略するとページにはスライドタイトルが表示されません)。また、frame 環境の外側では、\section{..} や \subsection{..} も通常の L^AT_EX ファイルのように使うことができ、\tableofcontents によって目次スライドを作成することができます (下の例でも使っています。参考にして下さい)。

beamer スライドソース例

```

\documentclass[dvipdfm]{beamer}
¥ usepackage[dvipdfmx]{hyperref}%hyper リンク
\usepackage{PXjahyper}% 日本語しおり
\usepackage{Madrid}% 他のテーマも試してみよう

\usepackage[english]{babel}%for English
\usepackage{amsmath,amssymb}%AMS 記号用
\usepackage{mathptmx}%math 用 Adobe Times Roman
\usepackage{helvet}%for normal english
\usepackage{courier}%\texttt{..}で courier
\usepackage[T1]{fontenc}%おまじない (1)
\usepackage{lmodern}%おまじない (2)
\usepackage{graphicx}% 各種画像用
% 箇条書きを段階的にみせたいとき
%\beamerdefaultoverlayspecification{<+>}

\title[Makin Slides using Beamer]
{\LaTeX{}}+Beamer でスライド作成}
\subtitle{\LaTeX{}}によるプレゼンテーション}

\author[Taro Meiji]{明治太郎}
\institute{明治大学理工学研究所}
\date[June 8 2013]{2013 年 6 月 8 日}
\subject{\LaTeX{}}+Beamer}

\begin{document}
\begin{frame}
\titlepage
\end{frame}

\begin{frame}<beamer>
\frametitle{Agenda}
\tableofcontents
\end{frame}

```

```

\section{はじめに}
\begin{frame}{何を問題としているか}
\begin{itemize}
\item こんなこと
\item あんなこと
\item しかも\alert{そんなことまで}
\end{itemize}
\end{frame}

\section{手始めに}
\subsection{Block の使い方}
\begin{frame}{さまざまな Block}
\begin{block}{ブロック}
これが block 環境だ。
\end{block}

\begin{example}
これは example block である。
\end{example}

\begin{alertblock}{警告ブロック}
alert block 環境ではこうなる。
\end{alertblock}
\end{frame}

\subsection{式を表示する}
\begin{frame}{\LaTeX{}}だから数式は得意だ}
Pauli 行列の性質は次のようだ。
\begin{equation}
[\sigma_x, \sigma_y]=2i\sigma_z,
[\sigma_y, \sigma_z]=2i\sigma_x,
[\sigma_z, \sigma_x]=2i\sigma_y
\end{equation}

```

ブロック環境でも数式を書ける

```
\vspace{0.5cm}
```

```
\begin{block}<+-->{内積の定義}
関数  $\phi(x)$  と  $\psi(x)$  の内積
```

```
\begin{equation}
\langle \phi, \psi \rangle =
\int \phi^*(x) \psi(x) dx
\end{equation}
```

```
\end{block}
```

```
\end{frame}
```

```
\section{図表の貼り込み}
```

```
\subsection{図}
```

```
\begin{frame}{PNG 画像}
```

PNG 形式の画像も、

bb ファイルを用意しておけば、この通り。

```
\begin{figure}
```

```
\includegraphics[scale=0.3]
{image/lorenz_flow.png}
```

```
\end{figure}
```

```
\end{frame}
```

```
\subsection{作表}
```

```
\begin{frame}{\LaTeX{}で作表してみる}
```

自分で\LaTeX{}コードで作表するのはチトきつい。

```
\begin{table}[htb]
```

```
\begin{center}
```

```
\begin{tabular}{l|r|l}
```

```
\hline
```

```
\multicolumn{3}{c}{シュークリームの材料}\\
```

```
\hline
```

```
\multicolumn{1}{c|}{品名} & 分量 & 用途\\
```

```
\hline
```

```
バター & 100g & シュー生地 \\
```

```
\cline{1-2}
```

```
塩 & 2g & \\
```

```
\cline{1-2}
```

```
小麦粉 & 100g & \\
```

```
\cline{1-2}
```

```
卵 & 3個 & \\
```

```
\hline
```

```
卵黄身 & 4個分 & カスタードクリーム\\
```

```
\cline{1-2}
```

```
砂糖 & 100g & \\
```

```
\cline{1-2}
```

```
小麦粉 & 50g & \\
```

```
\cline{1-2}
```

```
ミルク & 500cc & \\
```

```
\cline{1-2}
```

```
バニラエッセンス & 少々 & \\
```

```
\cline{1-2}
```

```
洋酒 & 少々 & \\
```

```
\hline
```

```
\end{tabular}
```

```
\end{center}
```

```
\caption{シュークリームのレシピ}
```

```
\label{tbl-cream}
```

```
\end{table}
```

```
\end{frame}
```

```
\section{結語}
```

```
\begin{frame}
```

```
\frametitle{わかったこと}
```

```
\begin{enumerate}
```

```
\item \LaTeX{}はとても便利
```

```
\item Beamer は Cool
```

```
\begin{itemize}
```

```
\item \LaTeX{}と beamer だけで
```

プレゼンテーションが可能

```
\item Nothing else?
```

```
\end{itemize}
```

```
\end{enumerate}
```

```
\vspace*{1.5cm}
```

```
\onslide
```

質問などは [\href{mailto:hogehoge@meiji.ac.jp}](mailto:hogehoge@meiji.ac.jp)

{\texttt{hogehoge@meiji.ac.jp}}にどうぞ

```
\end{frame}
```

```
\end{document}
```

10.2 Beamer でポスター

パッケージ `beamerposter`^{*3} は、Beamer 機能を拡張してポスターを作成するパッケージです (Beamer が使える環境が前提)。

次は `Beamerposter` の基本的な使い方の骨格を示しています。 `\documentclass[dvipdfm]{beamer}` とするのは通常の Beamer ファイルと同じです。次の行で、`Beamerposter` パッケージを用紙の使い方や A 版用紙サイズなどを指定して `\usepackage[orientation=portrait, size=a0, scale=1.4]{beamerposter}` のように読み込みます。用紙を縦置きにする場合は `orientation=portrait`、用紙を横置きにする場合には `landscape` とします。用紙サイズは `size=a0` として A4 から A0 まで選ぶことができます。

`\begin{document}` から始まる本文では、1 つだけの `frame` 環境が使われることに注意して下さい (Beamer では `frame` 環境ごとにページが生成されたことを思い起こして下さい。ポスターは 1 枚だけです)。この ‘大きな’ 1 つだけの `frame` 環境内に、`block` 環境 (`alertblock` も使えます) で記述したい事柄をブロック見出しを明記しながら記述するのです。

```
\documentclass[dvipdfm]{beamer}
\usepackage[orientation=portrait, size=a0, scale=1.4]{beamerposter}
\usetheme{使用するテーマ}
...
必要なパッケージ
...
\begin{document}
\begin{frame}
\begin{block}{見出し 1}
...Beamer スタイルで書く
\end{block}
\vfill
\begin{frame}
\begin{block}{見出し 2}
...Beamer スタイルで書く
\end{block}
....
....
\end{frame}
\end{document}
```

The `beamerposter` package に付属する `example.tex` を見てみると、用紙の向きに応じて段組を使って `block` 環境を使っています (`block` 環境は指定した幅を使い切りますから、用紙幅が大きければブロックは横に

^{*3} The `beamerposter` package <http://www.ctan.org/tex-archive/macros/latex/contrib/beamerposter>.

広がって縦に薄くなってしまい読みづらくなりますね)。次は、`block` 環境の幅をポスター幅 (`\linewidth`) の 0.48 倍として、ポスターの記述を 2 段組としている例です。

ポスター幅を 2 段で使う例

```
\begin{columns}[t]
\begin{column}{.48\linewidth}
\begin{block}{見出し}
...
\end{block}
.....
block 環境を適当回数繰り返す
.....
\end{column}
\begin{column}{.48\linewidth}
\begin{block}{見出し}
...
\end{block}
.....
block 環境を適当回数繰り返す
.....
\end{column}
```

`block` 環境が出現するたびに、左端段の上から下へ、右隣の段の上から下へと移動して表示されます。もちろん、1 枚のポスター内で 1 段組、2 段組、3 段組と組み合わせて利用することが可能です。The `beamerposter` package に付属するポスター例では、縦置きポスターで「1 段 +1 段 +2 段」で `block` を表示しています。一方、`RicePoster` パッケージ (Rice 大学用に `Beamer Poster` パッケージをカスタマイズ)^{*4} の例 `RicePosterExample.tex` では横置きポスターを 3 段組にし、真ん中の段を途中でさらに 2 段に分けています。

^{*4} LATEXresources for Rice students <http://ricebeamer.dynamaman.net>.

第 IV 部

さらに進む

第 11 章

文書作成再考

11.1 文書スタイルと組版エンジン

11.2 多言語文書の作成

付録 A

L^AT_EX 情報を調べる

付録 B

L^AT_EX のクラウドサービス

パソコン側の設定を一切することなく L^AT_EX 組版を実行し組版結果を PDF ファイルとして得ることができるクラウドサービスは Web ブラウザを通じて世界中どこからでもどんなデバイスからも利用できるのが重宝するので、登録しておく価値があります。次に代表的な L^AT_EX クラウドサービスを紹介します。どれも無料プランが用意されています。

- ShareLaTeX(<https://ja.sharelatex.com/>)
- CloudLaTeX(<https://cloudlatex.io/>)
- Overleaf(<https://www.overleaf.com/>)

これらには本書では説明しませんでした L^AT_EX 組版エンジンとして XeLaTeX を使う場合があります。XeLaTeX ではすべての文字は平たく UTF-8 で扱われるため日本語以外の文字言語を自由に混在して利用することが可能です。今日の標準的 T_EX インストール TeXLive を使ったシステム構築では必要なファイル全てが揃っているので、XeLaTeX に興味を持った方は調べてみるとよいでしょう (たとえば [T_EXWiki XeTeX](#))。

B.1 ShareLaTeX

チームで文書を共同作成せずに他の共同編集者がなく、Dropbox や GitHub との同期を必要としない個人利用は無料で利用できます。ShareLaTeX のアカウント登録は <https://ja.sharelatex.com/> からできます。

図 B.1 ShareLaTeX のログイン後の様子

ShareLaTeX では組版コンパイルエンジンに XeLaTeX を使っています。このため、多言語に対応したパッケージが必要です。日本語文書では文書クラスも `bxjsarticle` やパッケージ `ZXjatype` などを使わねばなりません。問題なく日本語 \LaTeX 文書の組版が可能です (ログインすると `japanese_doc` というプロジェクトで日本語文書の例が用意されています)。

図 B.2 ShareLaTeX の日本語文書作成

B.2 CloudLaTeX

CloudLaTeX は日本語 L^AT_EX 組版環境が大幅に改善された国産のクラウドサービスで、アカウントは <https://cloudlatex.io/> から登録できます。日本語文書で広く使われている組版コンパイルエンジンに pLaTeX (または uplatex) だけでなく、XeLaTeX も使うことができます (図 B.3 のように、そのテンプレートも多数用意されており便利です)。

図 B.3 ShareLaTeX のプロジェクト。[テンプレートから作成] からさまざまに用意されている雛形が使える。このように XeLaTeX 文書のサンプルも用意されている。

図 B.4 は、テンプレートで用意されている日本語レポートを組版している様子です。

図 B.4 CloudLaTeX の日本語レポートの作成の様子。[テンプレートから作成] の日本語レポートは図の貼り込みをする例だ。

B.3 Overleaf

付録 C

L^AT_EX のインストール

Windows や macOS のパソコンを持っている方はより身近に本格的な文書作成を行うためにも L^AT_EX をインストールしましょう。ここでは T_EX システムの配布形態として世界標準の TeX Live (sono macOS 版の MacTeX) を例に紹介します。

付録 D

L^AT_EX 処理の流れ

D.1 L^AT_EX システムの使い方

L^AT_EX システムをパソコンで利用するには、インストールした L^AT_EX システムを 1 つの作業空間で操作できる統合ソフトウェア TeXWorks(Windows/macOS) や TeXShop(macOS) などの L^AT_EX 統合ソフトウェアを使って L^AT_EX ファイルの作成、組版エラー処理、組版プレビューなどを行います (L^AT_EX システムのパソコンへのインストールは簡単で、75 ページの付録 C で紹介しています)。

L^AT_EX システムとその統合ソフトウェアがセットアップされていれば (L^AT_EX のクラウドサービスはそのような状態と考えることができますね)、L^AT_EX システム利用は組版コンパイル時のエラーメッセージに対処して組版された PDF ファイル生成することが主たる作業になります。L^AT_EX ファイル内のエラーを発見し、修正・保存して再コンパイルを繰り返すのです。

D.2 L^AT_EX システムの作業の流れ

L^AT_EX システムの詳細に立ち入る前に、L^AT_EX ファイルを作成し L^AT_EX システムを使って印刷するまでの手続きの流れを示します。

以下の手続きの流れはどんなコンピュータを使う場合でも同じです。TeX 統合環境を使えば、これらの大部分が隠蔽されて自動処理されるので、以下の TeX 作業は実際には難しくはありません。

1. 適当なテキストエディタを使って^{*1}、拡張子 .tex の付いたテキストファイル (L^AT_EX 文書ファイルとか TeX ファイルと呼びます) を L^AT_EX 規則にしたがって作成し、これを保存する。
2. TeX ファイルを コマンド platex によってコンパイルして、DVI ファイルの作成を試みる。
3. L^AT_EX システムからエラーが報告されたときには、処理を中断してエディタに戻り TeX ファイルを修正・保存してから再びコンパイルする。作業 2. と作業 3. をエラーがなくなるまで繰り返す。
4. 印刷する前に**プレビューア** (previewer) で DVI ファイルの印刷イメージを確かめる。TeX 統合環境を使うユーザは、コンパイルの後に生成された DVI から自動的に PDF ファイルが生成され、この作業をスキップできます。
5. dvipdfmx などのコマンドで DVI ファイルから PDF ファイルを生成してプレビューする^{*2}。

^{*1} Windows/Mac で利用できる TeXworks、Mac 専用の TeXShop や Windows 用に WinShell などの TeX 統合環境ソフトウェアはエディタ機能を内蔵していますが、手馴れたテキストエディタがあればそれも併用して使うことが可能です。

^{*2} DVI ファイルを経由せずにコマンド pdflatex などによって直接 pdf ファイルを生成することも可能ですが、日本語野処理に問

6. 生成した DVI/PDF ファイルを配布あるいは印刷する。

L^AT_EX システムにおけるこの処理の流れを図 D.1 に示します。

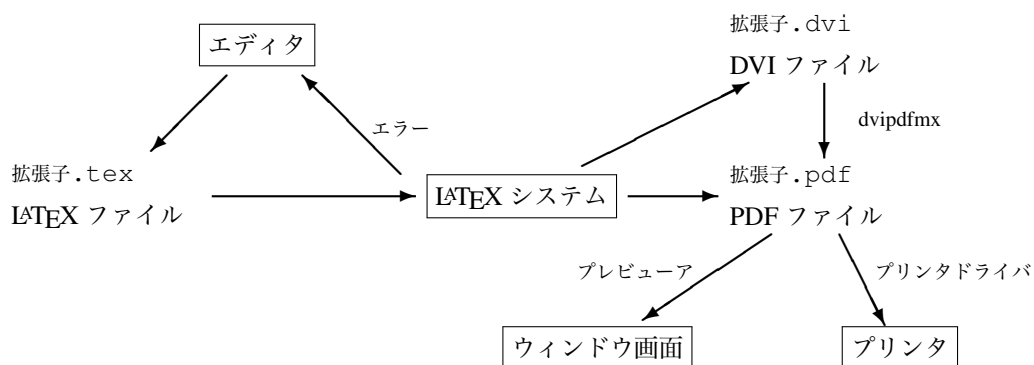


図 D.1 L^AT_EX システムにおける処理の流れ

TeX システムの利用者はワープロソフトウェアなどを使って文書を印刷する作業に比べて、図 D.1 のように ‘文法的’ に正しい L^AT_EX ファイルを作成して DVI ファイルを生成するという余計な手間を経ねばなりません。それであっても、TeX ユーザはさまざまなレイアウトで非常に美しい文書を作成できるという代償を得るのです。

D.2.1 DVI ファイル

DVI ファイルとは装置に独立 (DeVice Independent) な印刷イメージファイルで、使用しているコンピュータや印刷しようとするプリンタの種類とは無関係なバイナリファイルです。DVI ファイルには印刷に必要なフォントや各ページ上の座標位置などの情報が納められています。ただし、5.2 節で後述するように、DVI ファイルにはポストスクリプト画像ファイルの情報は含まれていません。画像情報を含むすべての情報を 1 つのファイルに収めるには、5.2 節で説明するように、dvipsk を使ってポストスクリプトファイルに変換するか、あるいは??節で触れるように dvipdfmx を使って PDF ファイルに変換する必要があります。

DVI ファイルが使用する装置に無関係であることは次の二つの意味があります。

- 使用したコンピュータに独立
Windows や Macintosh を使って T_EX システムから DVI ファイルさえ作成すれば、そのファイルを Linux に持ってきて (あるいは、その逆でも)、PDF ファイルを生成したり、プレビューしてプリンタから出力できます。
- 印刷しようとするプリンタに独立
あらかじめどのようなプリンタで印刷するかによらないで作業をすることができます。手元にある個人用のプリンタの出力結果と高価な印刷機からの出力結果の差は、その出力品質だけです。

題がある場合がある。

つまり、どんなコンピュータであろうとも DVI ファイル（さらに、それからポストスクリプトあるいは PDF ファイル）を作成してしまえば、そのファイルを高精度な印刷機を持つ印刷所に持ち込んで出力を頼めば、自分のパソコンで確認したレイアウトどおりの最高水準の印刷物が得られることになります。

D.2.2 PDF ファイル

PDF ファイルは Adobe 社によって規定されたページ出力のためのファイル形式です。PDF ファイルであれば無料配布されている Adobe Reader などを使ってどんなコンピュータでもその内容を表示できます。L^AT_EX システムでは、コマンド `dvi2pdfmx` を使って、次のようにして DVI ファイル `sample.dvi` から PDF ファイルを生成します。

```
% dvipdfmx sample.dvi
```

Adobe Reader が多くの PC での必携アプリケーションである今日の事情を考えると T_EX システムによって DVI ファイルではなく、PDF ファイルを最終生成ファイルとするのが今日の T_EX 利用の姿です。

D.3 組版出力までの手順

ここでは簡単な L^AT_EX 文書をエディタで作成し、これをプリンタから出力するまでの具体的手続きを紹介します。

節 D.3 以降の各段階の処理をコマンドによって行う様子を非常に面倒だと感じるかもしれません。多くの T_EX ユーザがそうであるように、統合環境を使えばプレビューまでの処理は自動化され、事実上 one click で済んでしまいます。ただし、高度な T_EX 機能を利用する場合には、ここで説明するようなコマンドによる処理が必要になる場合もあります。

Windows では [アクセサリ]-[コマンドプロンプト] からコマンドプロンプトウィンドウ（俗称 DOS 窓）を、Mac では [アプリケーション内のユーティリティにあるターミナル app] を実行して、コンピュータに実行させる命令（コマンド）を文字列として入力することができます。図 D.1 の処理の流れさえ把握していれば、OS の違いによる混乱は起こらないでしょう。

L^AT_EX 文書の作成の実際

テキストエディタを使って作成する L^AT_EX ファイルは常に `.tex` という拡張子を付ける必要があります。簡単な L^AT_EX ファイルの例として、次のように入力したファイルを `sample.tex` として保存してみましょう（実はワザと誤りを入れてあります）。

```
\documentclass{jsarticle}% この jsarticle が現在の事実上の標準である
\begin{document}          % 全角文字『』を使った誤り
初めての\TeX{}文書です。
どんなふうに
仕上がるかな?

とにかくエディタで文書さえ書いておけば、後で\LaTeX{}ファイルにすることは簡単。
```

```
やっぱり文章は見てくれより{\Large 中身}
が大切だからね。
```

でも、こんなこと

```
\[
\int_0^{\infty} \frac{\sin ax}{x} dx=
\frac{\pi}{2}\quad (a>0)
```

```
\]
```

が書けるとなると、`\textbf{外見も大切}`かなあとってしまうよね。

```
\end{document}
```

ここで表示されている半角のバックスラッシュ記号 ‘\’ について一言。Windows や Macintosh などの日本語環境では通常この半角バックスラッシュ記号は Yen キーで入力され、¥ というように表示されるものです。以下では、自分の環境に合わせて ‘\’ を ‘¥’ と読み替えて下さい。

この簡単な例は、**L^AT_EX 文書ファイルの必須要素**を示しています。

- まず `\documentclass{jsarticle}` を書いて日本語論文クラスを指定する。jsarticle は標準的な日本語論文を組版します*3 (出版社や学会から様々な文書スタイルが提供されています)。
- 文書スタイルの指定の下に `\begin{document}` を書く。`\documentclass{...}` と `\begin{document}` の間を行間部分を **プリアンプル**といい、通常はさまざまな情報を書きま
す (9 ページの 2.1 節以降で紹介していきます)。
- 文書本体は `\begin{document}` の後から書き始める。
- 本文の最後に `\end{document}` を書く。

なお、ファイル中の記号 ‘%’ は、それ以降から行末までを**コメント**として扱うために使われています (D.3.1 節)。コメントは仕上がり文書には反映されませんが、メモや修正事項などを記入したり L^AT_EX 文書を分割して個別の文書ファイルのデバッグをするなどさまざまな用途があり、たいへん便利です (D.3.1 節)。

DVI ファイルの作成

保存した L^AT_EX ファイル *sample.tex* を L^AT_EX システムによって**コンパイル**して DVI ファイル *sample.dvi* を作成するには次のようにします。

```
% platex sample.tex
```

このとき L^AT_EX 構文上のエラーがあるとき、L^AT_EX システムは途中で処理を中断し、違反あるいは処理が破綻したファイルの行番号をエラーメッセージとともに示し、“?” のプロンプトによって利用者にこれ以降の処理を尋ねてきます。実際には、その箇所でエラーが起こったというよりも、それ以前にあった誤入力によって矛

*3 この jsarticle は以前の jarticle を奥村晴彦氏が改良した日本語用論文スタイルで、L^AT_EX の日本語文書クラスに標準的に含まれています。

盾が積み重なってエラーとして顕在化する場合が多く、エラーの修正にはその行以前にさかのぼって探査する必要があります。

“?”のプロンプトは、このエラーをユーザがどのように対処するかを L^AT_EX システムに指示するために、キーボードからのコマンド“入力待ち状態”を表しています。上の `sample.tex` の場合では次のようなエラー状態となります（以下の例は、MacOS で MacTeX を利用した場合です）。この例では、`\begin{document}` のように、半角の ‘}’ でなければならないのに、日本語モードで入力した全角の ‘}’ が使われてしまったためにエラーが発生しました。

```
... 利用している OS や使っている TeX システムに応じたメッセージが続く
This is e-pTeX, Version 3.1415926-p3.3-110825-2.4 (utf8.euc) (TeX Live 2012)
restricted \write18 enabled.
(./test.tex
pLaTeX2e <2006/11/10> (based on LaTeX2e <2011/06/27> patch level 0)
.....
(/usr/local/texlive/2012/texmf-dist/tex/platex/jsclasses/jsarticle.cls
Document Class: jsarticle 2010/03/14 okumura
)
Runaway argument?
{document}  どのなふうに仕上がるかな?
! Paragraph ended before \begin was complete.
<to be read again>
                \par
1.5
? ■ ← コマンド入力待ち
```

処理中のすべてのメッセージは、拡張子 `.log` のついたログファイル `sample.log` として自動的に保存されますから必要に応じて随時参照します。

このようにエラーが原因で処理がとまってしまった場合、L^AT_EX システムに入力できる代表的な対処コマンドキーには次のものがあります。

- h エラー原因をアドバイスしてくれる（あまり役に立たない）
- x 処理を強制終了する（処理を中止してエディタで修正する場合は、このコマンドを使います）
- q エラーや警告を無視して処理を実行してしまう
- ? コマンド一覧の表示
- エラーに構わず次の処理を行なう

日本語の全角記号や全角空白入力によって生じるエラーは少なくありません。そのような日本語独特の課題によって生じるエラーを発見しやすくするためには、半角空白文字や全角空白文字を区別して表示をさせるといったテキストエディタ機能を存分に利用するとよいでしょう。

エラーがあったとき、通常は ‘x’ を入力して L^AT_EX 処理を中断し、起動してあるエディタ画面に戻って修正・保存します。エラー箇所を修正したファイルを保存して、それを再びコンパイルして DVI ファイル作成を試みます。こうして L^AT_EX 構文エラーがなくなるまで以上の作業を繰り返します。最終的には L^AT_EX システムから次のようなメッセージ（以下の例は、MacOS で MacTeX を利用した場合）。

```
... 利用している OS や使っている TeX システムに応じたメッセージが表示される
This is e-pTeX, Version 3.1415926-p3.3-110825-2.4 (utf8.euc) (TeX Live 2012)
restricted \write18 enabled.

.....
(./test.tex
pLaTeX2e <2006/11/10> (based on LaTeX2e <2011/06/27> patch level 0)
.....
(/usr/local/texlive/2012/texmf-dist/tex/platex/jsclasses/jsarticle.cls
Document Class: jsarticle 2010/03/14 okumura
) (./test.aux) [1] (./test.aux)
Output written on test.dvi (1 page, 1028 bytes).
Transcript written on test.log.
```

が表示され、DVI ファイル `sample.dvi` が生成されたことがわかります（確認するにはどうすればよいかわかりますか）。この例では短い文ですから 1 ページしかありませんが、長い文のときには [1] [2] [3]… とページ数が増えていきます。

DVI ファイルのプレビュー

現在の L^AT_EX システムではプレビューするには 3 通りの方法があります。1 つ目は DVI ファイルをプレビューするソフトウェアを使うやり方と、2 つ目は DVI ファイルを PostScript ファイルに変換 (??節) してからプレビュー・印刷するやり方、3 つ目は DVI ファイルを PDF ファイルに変換 (??節) してからプレビュー・印刷するやり方です。

多くの TeX ユーザはエディタ機能を備えた統合環境を利用したり、エラー処理、プレビューおよび修正作業を自動化するスクリプトを利用して、今日では PDF ファイルを生成してプレビューする 3 つ目の方法が標準的となっています。したがって、以下の DVI ファイルのプレビューに関する記述は事実上不要でしょう。実際、L^AT_EX 利用者や Linux 利用者以外のユーザに DVI ファイルや PostScript ファイルメールに添付してプレビューできるシステム環境を整えている利用者は多くはありません。

生成された DVI ファイル `sample.dvi` ファイルからいきなり印刷せずに、プレビューアを使って印刷イメージを確認します。作成された DVI ファイルのプレビューは、Windows では「dviout for Windows」が有名です。Linux ではプレビューアとして `xdvi` を使って DVI ファイルの印刷イメージを次のようにして確かめることができます。

```
% xdvi sample.dvi
```

5.2 節で述べるように、PostScript 画像（正確には Encapsulated PostScript ファイル）が張り込まれている DVI ファイルを画像といっしょにプレビューするためには Ghostscript のインストールが必要です。節??で紹介し

た TeX システムのインストールに従ったならば、既にインストールされているはずです。

DVI ファイルの印刷

プレビューで印刷イメージを確認し終わって、それが期待どおりであればプリンタに印刷します。

Windows ではプレビューア dviout for Windows、Mac では xdvi などの DVI プレビューアから直接印刷できます。Linux などからポストスクリプトプリンタを使う場合には、DVI ファイルを Postscript コードに変換する dvipsk (または dvips) を使って次のようにして印刷することができます。

```
% dvipsk sample.dvi
```

今日の TeX 作業のゴールは Adobe Reader でプレビューすることができる PDF ファイルの生成です。PDF ファイルは DVI ファイルから次のコマンドで生成します (??節)。

```
% dvi2pdf sample.dvi
```

プレビューすると、次のような結果が得られるでしょう。

初めての TeX 文書です。どんなふうになるかな?
とにかくエディタで文書さえ書いておけば、後で LaTeX ファイルにすることは簡単。
やっぱり文章は見てくれより **中身** が大切だからね。
でも、こんなこと

$$\int_0^{\infty} \frac{\sin ax}{x} dx = \frac{\pi}{2} \quad (a > 0)$$

が書けるとなると、**外見も大切**かなあと思ってしまうよね。

D.3.1 LaTeX ファイルと表示

LaTeX コマンド

78 ページの LaTeX ファイル例には、たとえば ‘\LaTeX{}’ というような特別な書き方を含んでいます。DVI ファイルを作成して印刷するとこれは ‘LaTeX’ のように印刷されます。

このように LaTeX ファイルでは、**バックスラッシュ**記号 ‘\’ に続いたある特別な文字列を使って特別な印刷結果や組版上の効果を得ることがあります。これらを **LaTeX コマンド**といい、決められた書き方をしなければなりません。LaTeX システムが前もって定めていて勝手には変更できない言葉を **予約語**といいます。

たとえば、‘\TeX{}’ とすると ‘TeX’ という結果になります。‘\TeX’ はコマンドとして予約されていますが、‘\tex’ や ‘\Tex’ はコマンドとして予約されていないので LaTeX システムはエラーを出します。また、“\LaTeX **文書は美しい**” のように予約語 ‘\LaTeX’ に続けて他の言葉を書くと、‘\LaTeX **文書**’ などが LaTeX コマンドとしてシステムに解釈されることとなります。しかし、そのようなコマンドは LaTeX システムには登録されていないのでエラーとなります。

LaTeX ファイル中の文字列のどこが LaTeX コマンドであるかを明示するために、次のような工夫をすればよいでしょう。

- ‘\LaTeX{} **文書は美しい**’ のようにコマンドの直後に ‘{}’ を追加する

- ‘\LaTeX_文書は美しい’ のようにコマンドの直後に半角空白を挿入する
- ‘{\LaTeX}文書は美しい’ のようにコマンドの有効範囲を定めるために ‘{’ と ‘}’ で囲んで ‘\LaTeX’ がコマンドであると明示する。

コメントの活用

LaTeX では記号 ‘%’ 以降から行末までコメントとして扱われます。不要となった文章やファイルに関する補助情報などをコメントにしておくことで後で活用できます。また、コメント機能には LaTeX ファイルの分割 (16 ページ) を使って文書の内容を制御する大切な役割があります。

ここで、行末というのはエディタ上で Return キー (あるいは Enter キー) を押して改行記号を入力した箇所を意味します。エディタから見れば、改行記号から改行記号までの文字列が論理的な 1 行とみなされます。

改行と段落の取り扱い

上の例で、エディタ上での LaTeX ファイル内容とその印刷結果 (またはプレビュー画面) を見比べてわかるように、LaTeX ファイルでの改行と印刷出力での対応は次のようになっています。

- LaTeX ファイル内の単なる改行は、印刷出力では改行とはならない。文章途中でいくら改行しても、同じ段落を構成する一連の文章とみなされる。
- 印刷出力において段落を改めるためには、LaTeX ファイルで一つ以上の空行 (行頭で改行すること) を入れるか、または行末でコマンド ‘\par’ を記入する。このとき、段落始めとなる文章が行頭から書いてあっても、出力されると適当にインデント (字下げ) が行なわれる。
- LaTeX ファイル内で行末にコマンド ‘\’ を記入すると、出力では強制改行される。このとき、改行後の行頭文字はインデントされない。

このように LaTeX システムでは、段落と段落の区切りには (一つ以上の) 空行またはコマンド ‘\par’ がその役目を果たします。したがって、LaTeX ファイルの単なる改行では段落が改まらないことを利用して、エディタで LaTeX ファイルを作成するときには「1 文で改行して 1 行とする」ように書くとテキスト編集の能率が向上するでしょう。

半角文字と全角文字

日本語モードで入力した文字を**全角文字**といい、テキスト画面上では下の表のように全角文字は半角文字の倍の文字幅となっています。

半角文字	{abcdefg}
全角文字	{a b c d e f g}

コンピュータにとっては全角文字と半角文字は異なる文字として識別されています。したがって、エディタを使っているときに半角文字か全角文字のどちらの種類の文字を入力しているかを常に意識しておかなければなりません。

LaTeX ファイルでは、上の例でエラーの原因となったような ‘\’ や ‘{’, ‘}’ などの特殊文字や記号はすべて半角文字で書かなければなりません。特に空白文字には気を付けます。モニタに表示される時、半角空白

‘ ’と全角空白‘ ’は区別が付きにくいからです。半角空白文字と全角空白文字を区別して表示できるようなエディタを使うとよいでしょう。

付録 E

Web ページで L^AT_EX 記法を使う

参考文献

- [1] 奥村晴彦・黒木 裕介, 『[改訂版第 3 版]L^AT_EX2_ε-美文書作成入門』, 技術評論社 (2010 年) .
- [2] 吉永徹美, 『LaTeX2e 辞典 用法・用例逆引きリファレンス』, 翔泳社 (2009).
- [3] 監修:渡辺徹, 『L^AT_EX による論文作成の手引き』 (2006), [http://mytexpert.osdn.jp/?LaTeX による論文作成の手引き](http://mytexpert.osdn.jp/?LaTeX%20による論文作成の手引き).
- [4] T_EX Wiki, <https://texwiki.texjp.org>.
- [5] F. Mittelbach, M. Goossens, et al., ‘*The LaTeX Companion* (2nd Edition)’, Addison-Wesley(2004). [邦訳] アスキー編集部監訳, 『The LaTeX コンパニオン』 (第 1 版) , アスキー出版局 (1998 年).
- [6] Donald E. Knuth, 『改訂新版 T_EX ブック—コンピュータによる組版システム』, アスキー出版局 (1992 年) .
- [7] L. ランポート, 『文書処理システム L^AT_EX2_ε』, ピアソンエデュケーション (1999).