

2004 年度版 改訂 第 2 版 (誤植訂正版)

公立はこだて未来大学
システム情報科学部
FUNNIST 編集委員会

監修：渡辺徹

Copyright © 2003, 2004 FUNNIST

Copyright © 2003, 2004 渡辺徹

この文書をフリーソフトウェア財団発行の『GNU フリー文書利用許諾契約書』（バージョン 1.1 かそれ以降から一つを選択）が定める条件の下で複製，頒布，あるいは改変することを許可する．変更不可部分，表カパーテキスト，裏カパーテキストは指定しない．この利用許諾契約書の複製物は *GNU Free Documentation License* という章（付録 A）に含まれている．日本語訳『GNU フリー文書利用許諾契約書』は非公式なものであり法的有効性はないが，

<http://www.gnu.org/copyleft/fdl.ja.html>

で参照可能である．

本冊子に記載されている企業，団体の名前や製品名等はそれぞれの権利帰属者の商標または商標登録であり所有物です．本冊子では™及び®は明記していません．

まえがき

これは何のための冊子か

レポート・論文を執筆するためには、まず何を書くのかを決めます。しかし、書くべき内容が決まったとしても、「どのように」書けば良い論文になるのかは不明瞭な部分があります。

この冊子では「どのように」論文を執筆すべきかを解説し、文献の探し方や書くべき内容には言及しません。

この「どのように」に答える一つの方法として^{らてっく}LaTeXと呼ばれるプログラムを用いることが考えられます。LaTeX は科学技術系の論文の作成などに広く使われています。さらにマークアップ方式を採用しているので、原稿の汎用性が比較的高くなります。

LaTeX の導入やその周辺の情報に関しては、三重大学の奥村晴彦氏のウェブページ

<http://oku.edu.mie-u.ac.jp/~okumura/texwiki/>

等を参照してください。

また、ただ眺めるのではなく、実際に自分で入力し、実行結果を吟味してみることをお勧めします。

この冊子に対して追加、削除、変更すべき事項などがあればお知らせ下さい。特にこの冊子の分かりづらい箇所、もっとうまく説明すべき箇所などについて連絡をお願いします。連絡先は奥付を参照してください。

FUNNIST について

このような冊子の作成をしている組織に名前があります、これを *Future University-Hakodate Network and Information System Tutorial* と呼んでおります。公立はこだて未来大学を英語では *Future Univeristy-Hakodate* と省略して呼ばれますので *FUN* となります。この冊子を作成している組織は未来大学の情報システムやネットワークシステム、果てはコンピュータの総合的な使い方を示す手引書 (*Tutorial*) を作成することを目的としています。

この組織での重要な対象はネットワークと情報システムの二つです。この二つを英語にすると *Network* と *Information System* です。そのような理由もあり、この組織の名称は *Future University-Hakodate Network and Information System Tutorial* に決定しました。省略形として FUNNIST も正式なものとして用いることが出来ます。

凡例

本冊子では書体を変更することによって同じ語句でも違った意味を持つものが多数あります．‘dvipdfm’ という語があったとしても ‘dvipdfm’ や ‘dvipdfm’ , ‘dvipdfm’ , ‘dvipdfm’ はすべて別の意味を持っています．これらの書体の種類については 3.19 節を参照してください．

書体	意味	例
ローマン体	通常の記事	dvipdfm
サンセリフ体	パッケージやクラス (3.21 節参照)	dvipdfm
タイプライタ体	キーボードからの入力など	dvipdfm
イタリック体	変数や強調	<i>dvipdfm</i>
スラント体	オプション (3.21.2 節参照)	<i>dvipdfm</i>

本文中で左側にタイプライタ体，右側にそれに準じた出力例があるものは，入出力の対を表します．

The length of a pen should be comrotable to write with: too long and it makes him tired; too short and it\ldots.	The length of a pen should be comrotable to write with: too long and it makes him tired; too short and it...
--	--

テキストエディッタなどを使い，原稿ファイルで左側のように入力すると，右側の出力例と同じような結果を確認できます．

文中において `which perl` という表記はコマンドプロンプトやシェルなどのコンソールからの入力を示します．複数行の入力の場合は次のようにしています．

```
$ platex file.tex
$ jbibtex file.tex
$ dvipdfmx -S -o out.pdf input.dvi
```

先頭のドル ‘\$’ はコンソールに表示されている記号で，ユーザは入力しません．

キーボード上の特定のキートップを押すことを示すには `[Alt]` のようにします．`[Ctrl]+[Alt]+[Delete]` は `[Ctrl]` , `[Alt]` , `[Delete]` キーを同時に押すことになります．`[Ctrl]+[x]` `[Ctrl]+[s]` は `[Ctrl]+[x]` を押した後に `[Ctrl]+[s]` を押すことを表します．

何らかの文字列や数値に置き換わるものは `<変数>` のように表記しています．

謝辞

この冊子を作成するためには非常に多くの方々のご協力，ご助言がなければ実現することが難しかったことを容易に想像できます． \LaTeX 全般に関しては秋田純一氏，奥村晴彦氏，吉永徹美氏より多くのことを学びました．

\TeX の作者である Donald Knuth 氏， \LaTeX の作者である Leslie Lamport 氏， $\text{\LaTeX} 2_{\epsilon}$ の開発をされた Frank Mittelbach 氏，Johannes Braams 氏，David Carlisle 氏，Michael Downes 氏，Alan Jeffery 氏，Sebastian Rahtz 氏，Chris Rowley 氏，Rainer Schöpf 氏， \TeX の日本語化をして下さった中野賢氏とアスキーの方々，Windows に $\text{p}\text{\TeX}$ を移植してくださった角藤^{あきら} 亮 氏， dviout を開発された大島利夫氏と乙部^{よしき} 徹己氏， $\text{Bib}\text{\TeX}$ の開発をされた Oren Patashnik 氏， MakeIndex を開発・改良された Pehong Chen 氏と Nelson Beebe 氏， dvipdfm の作者である Mark Wicks 氏， Dvipdfmx の保守・管理をされておられる平田俊作氏と Cho Jin-Hwan 氏，PostScript や PDF などのページ記述言語を作成された Adobe 社の方々，フリーウェア・マクロパッケージなどの作成で， \TeX の分野において貢献された方々にも感謝いたします．

さらに，この冊子を直接ご覧いただき，ご助言を戴いた大友康寛氏，田中健太氏，永田善久氏，野村昌孝氏，三上貞芳氏，和田志保美氏らには感謝しております．執筆段階での不備や不具合を指摘していただき，この冊子の作成が円滑に進み，また主観的になりすぎていた部分も見直すことが出来ました．

最後に，この冊子を作成・発行することを快く進めてくださった木村健一先生には深く感謝の意を表したいと思います．木村先生のお陰で，この冊子が陽の目を見たといっても過言ではありません．

変更履歴

この冊子は渡辺徹氏が L^AT_EX の入門書として配布した『好き好き L^AT_EX 2_ε 初級編』を論文執筆に沿うように、FUNNIST が大幅に改変した文書です。『好き好き L^AT_EX 2_ε 初級編』は 2004 年 4 月 2 日に α 版を公開し、現在は 250 ページを超える大規模な入門書となっています。もしも、この冊子で不十分な部分があれば、下記ウェブページで配布している各種 L^AT_EX に関する入門書・専門書を参照してください。

<http://tex.dante.jp/>

2003 年 11 月に『L^AT_EX による論文作成の手引き』という名前で第 1 版を配布しました。その後、より一般的な用途に使用できるように、L^AT_EX の入門書を 2004 年 4 月 2 日に配布しました。さらに、入門書をベースに FUNNIST が『L^AT_EX による論文作成の手引き』を書き直しました。

2003 年 11 月

『L^AT_EX による論文作成の手引き』という名前で、主に大学内で冊子を配布する。

2004 年 4 月 2 日

論文作成にとらわれない入門書として、上記手引きを基盤に『好き好き L^AT_EX 2_ε 初級編』 version 0.1 を配布する。

2004 年 4 月 16 日

2004 年 4 月 2 日の誤記・誤植を修正し version 0.2 を配布する。

2004 年 4 月 30 日

2004 年 4 月 16 日でも発見されなかった誤記・誤植情報を修正し、version 0.2a を配布する。

2004 年 8 月 5 日

あまりに肥大化した冊子において、入門書が必要としないと思われる部分（拡張的な記号など）を削除した version 0.3 を配布する。

2004 年 8 月 19 日

「参考資料」の章の体裁が変則的になっていた部分を改変し、version 0.3a を配布する。

2004 年 9 月

上記入門書の簡易版であり、論文作成に特化し、題名を改め『L^AT_EX による論文作成の手引き』を配布する。これは上記入門書の派生物であり、全くの別物という位置付けで発行した。論文作成に直接的に必要なと思われる部分を削除した。

2005 年 3 月 20 日

2004 年 9 月に発行した第 2 版の誤植訂正版を作成し、これを配布しました。

目次

まえがき	i
------	---

謝辞	iii
----	-----

第 1 章 論文作成を始める前に	1
------------------	---

1.1 論文とは何か	1
1.2 組版とはなんだろうか	1
1.3 文章表現	1
1.4 T _E X とは何か	3
1.5 WYSIWYG とは何か	3
1.6 一括処理とは何か	3
1.7 L ^A T _E X とは何か	3
1.8 情報の入手先	4

第 2 章 L ^A T _E X の基本	5
---	---

2.1 基本の基本	5
2.1.1 処理の流れ	5
2.1.2 動かしてみる	6
2.1.3 原稿作成時の注意点	8
2.1.4 フォルダ・ファイルの基本的な操作	8
2.1.5 エラーに遭遇する	10
2.1.6 プレビューアの操作	12
2.1.7 コマンド	14
2.1.8 括弧について	15
2.2 L ^A T _E X に関わるファイル形式	15
2.3 コマンドの基本	16
2.3.1 プリアンブルでのコマンド	17

第 3 章	文章の書き方	19
3.1	文章の論理構造	19
3.2	表題	20
3.3	見出し	21
3.3.1	見出しの出力	21
3.3.2	見出しの深さ	21
3.4	目次の出力	22
3.4.1	目次を出力する深さ	22
3.4.2	見出しの番号付けの深さ	22
3.5	概要の出力	23
3.6	段落と字下げ	23
3.6.1	行頭の下げの有無	24
3.6.2	ダブルスペース	25
3.7	長さの単位	25
3.7.1	L ^A T _E X での単位の取り決め	25
3.7.2	単位の使い方	26
3.8	句読点	26
3.9	注釈	27
3.10	文字の強調	27
3.11	特殊記号	28
3.12	原稿中での空白の扱い	28
3.13	コメントの挿入	29
3.14	べた書き	29
3.15	引用や文の区切り	30
3.15.1	書籍名や雑誌名の引用	31
3.15.2	ダッシュ	31
3.15.3	改行	32
3.16	空白について	33
3.16.1	文章の中の空きの調整	33
3.16.2	その他注意すること	34
3.16.3	和文と欧文のあいだの空白	34
3.17	箇条書き	35
3.18	行揃え	38
3.19	書体について	39
3.19.1	文字の大きさの変更	39
3.19.2	書体の変更	41
3.20	文章の修正	42
3.21	クラスとパッケージ	43

3.21.1 標準的なクラス	43
3.21.2 クラスオプション	44
3.21.3 標準で利用できるパッケージ	45

第 4 章 参考文献の出力 47

4.1 参考文献の明記	47
4.2 参考文献を手動で並べる場合	48
4.2.1 文献の並べ方	49
4.3 参考文献をプログラムで並べ替えるとき	50
4.3.1 jBibTeX の使い方	50
4.3.2 文献データベースの作成	50
4.3.3 参考文献一覧の出力	51
4.3.4 文献の種類及び項目	52
4.3.5 各文献スタイルの出力例	55
4.3.6 文献の追加例	55
4.3.7 文献を同時に複数参照しているとき	57
4.3.8 参照の形式を変更する	57
4.4 文献一覧の管理	58

第 5 章 原稿の出力形式 59

5.1 出力形式の種類	59
5.1.1 DVI	60
5.1.2 PostScript	60
5.1.3 PDF	60
5.2 DVI を PS に dvipsk	61
5.3 DVI を PDF に Dvipdfmx	62
5.4 画像ファイルの扱い	63

第 6 章 コマンドとマークアップ 65

6.1 マークアップ言語とは?	65
6.2 記号とコマンド	65
6.2.1 記号の分類	65
6.2.2 コマンド	66
6.2.3 コマンドの定義	67
6.2.4 文字やコマンドの区切り	68
6.2.5 コマンドの引数	70
6.3 グループ・入れ子構造	71

6.4	宣言と命令の違い	71
6.5	相互参照	73
6.5.1	相互参照の仕組み	73
6.5.2	カウンタ	74
6.6	相互参照の工夫	75
6.6.1	相互参照に関わる L ^A T _E X の警告	78

第 7 章 数式の書き方 79

7.1	はじめに	79
7.2	数式の出力	79
7.2.1	文中数式	79
7.2.2	グルーピング	80
7.2.3	別行数式	80
7.2.4	番号付き数式	81
7.2.5	複数行数式	81
7.2.6	複数行番号付き数式	81
7.3	書体の変更	82
7.4	数式における空白の調節	83
7.5	基本的な数式コマンド	83
7.5.1	添え字	83
7.5.2	数学関数	84
7.5.3	大きさ可変の数学記号	85
7.5.4	区切り記号と括弧	86
7.5.5	行列	88
7.6	表示形式の調整	90
7.7	数式モード中の記号	91
7.7.1	ギリシャ文字	91
7.7.2	関係子や演算子などの数学記号	93
7.7.3	標準ではない数学記号	95
7.8	定義や定理など	95
7.8.1	定理型環境のカスタマイズ	96
7.9	雑多なこと	97
7.9.1	記号の積み重ね	98
7.9.2	記号の重ね合わせ	99
7.9.3	数式の太字	99
7.9.4	高さを揃える	100
7.9.5	スマートな分数の書き方	101
7.9.6	数式モード中の空白と書体	101

第 8 章 図表の貼り方 103

8.1	図表の基礎	103
8.1.1	一般的な取り決め	103
8.1.2	L ^A T _E X での扱い	104
8.2	表	105
8.2.1	表中の脚注	107
8.2.2	表作成支援ツール	107
8.3	図	108
8.3.1	EPS 画像の張り込み	108
8.3.2	他のプログラムから張り込む方法その 1	112
8.3.3	他のプログラムから取り込むとき	113
8.3.4	図を二つ横に並べる	114
8.3.5	EPS 以外の画像の張り込み	114
8.4	描画の方法	115
8.4.1	化学式・化学構造式	115
8.4.2	グラフの描画	116

第 9 章 L^AT_EX の応用 117

9.1	レイアウトの制御	117
9.2	あらかじめ定義されている見出しの変更	118
9.3	多段組	118
9.4	箱の操作	119
9.4.1	枠のない箱	119
9.4.2	枠のある箱	120
9.4.3	広範囲な箱	120
9.4.4	罫線と下線	121
9.5	空白の挿入	121
9.5.1	水平方向の空き	122
9.5.2	垂直方向の空き	123
9.6	原稿を複数のファイルに分ける	123
9.7	用語の統一	124
9.8	URL の記述	125
9.9	原稿の執筆支援	125
9.9.1	入力支援統合環境 YaTeX	125
9.9.2	入力支援統合環境 WinShell	126
9.9.3	latexmk	126

第 10 章 論文のサンプル	127
10.1 中間報告のサンプル	127
10.2 学位論文のサンプル	133
10.2.1 クラスファイルが提供されている	133
10.2.2 クラスファイルが提供されていない	135
付録 A GNU Free Documentation License	139
1. APPLICABILITY AND DEFINITIONS	139
2. VERBATIM COPYING	141
3. COPYING IN QUANTITY	141
4. MODIFICATIONS	142
5. COMBINING DOCUMENTS	143
6. COLLECTIONS OF DOCUMENTS	143
7. AGGREGATION WITH INDEPENDENT WORKS	144
8. TRANSLATION	144
9. TERMINATION	144
10. FUTURE REVISIONS OF THIS LICENSE	144
ADDENDUM: How to use this License for your documents	145
参考文献	147
索引	149

第 1 章

論文作成を始める前に

この章では L^AT_EX 使用の前提知識を紹介します。L^AT_EX はマークアップ式の文書整形プログラムですから、それに関わる知識や文書の構造について知るのには効果的だと思われます。

1.1 論文とは何か

大学 3 年間で学んだことを生かして研究をし、考察する文書のことです。自分の書いた論文は自分で眺めるために書くわけではないと思います。より多くの人に自分の研究成果を伝えるために、作成するものです。その観点から言えば、論文というのは書き殴りの様な文体では読み手に理解してもらえません。さらに体裁も整っていないければ、読み手の理解力を低下させることになります。正しい記述で、読者にとって見やすい紙面構成を心がけ、かつ内容も独自性が高いと、良い論文に仕上がることになります。

1.2 組版とはなんだろうか

くみはん
組版とはある媒体、特に書籍などの紙のうえに読者が読みやすいように必要な情報を適切な位置に配置することです。

現代ではコンピュータ上で文書を組版できるようになりました。だれでも手軽に印刷用の美しいフォントを用いた組版が可能です。ここで文書がどのようにして組版されているのかを少し説明します。

世界中で出版されている書籍・論文などは一定のルールに沿って整形されているものです。たとえば 1 行を何文字にするか 1 ページを何行にするかなどの約束事があります。このような様式をどのようにするのは各出版社や各種学会が一定の条件を判断して決めています。

なぜこのような決まり事があるかという文字や図を含む本や雑誌は必ず誰かに見てもらう、読者を相手にしていることを前提としているからです。その本の内容に合わせて読者にとって読みやすい本とは何かを追求してこのような様々な書式が存在します。

1.3 文章表現

L^AT_EX を用いるとユーザーがそのような高度な技術を持っていなくてもプログラムが半

自動的に組版するようになっていきます。しかし最低限のルールを覚えなければ、とても出たら目な文書に仕上がってしまいます。

次の例文の中には多くの文章表記の上での約束事が秘められています。

The length of a pen should be comrotable
to write with: too long and it makes him
tired; too short and it\ldots.\par
When I was a young---a foolish boy---the
pen was too long! So I used to break it.

The length of a pen should be comrotable to
write with: too long and it makes him tired;
too short and it...
When I was a young—a foolish boy—the pen
was too long! So I used to break it.

ここでは句読点とダッシュの用法が確認できます。コロンの、セミコロンなどの記号はコンマ、ピリオドと同様に、記号の前に空白（空き）を入れず、後ろに半角の空白を挿入しています。文を中断するダッシュ^{エムダッシュ}、em-dash の場合は前後に空白を入れません。

“\, ‘Stop!’ the man said.” \par
Prof.~Albert Einstein (1897--1955)
was born in German (see fig.~3).\par
His famous equation \$ E = mc^2 \$ is
written in the theory.

“‘Stop!’ the man said.”
Prof. Albert Einstein (1897–1955) was born in
German (see fig. 3).
His famous equation $E = mc^2$ is written in the
theory.

クオートで一文を引用していますが、引用の中の引用とクオートが隣接している部分は若干の空白を挿入しています。アインシュタインが 1897 年から 1955 年まで生きていたという、数値の範囲を示す場合は^{エンダッシュ}en-dash を用います。日本語でも波ダッシュ‘~’は使いません。「図 3 を参照せよ」という意味の‘(see fig. 3)’ですが、丸括弧（パーレン）の左側（起こし）に空白を入れています。右側（受け）には入れていません。‘fig.’ と ‘3’ の間で改行することは好ましくないため、チルダ‘~’を補っています。数式中の等号‘=’は関係演算子を意味していますので、前後に適切な空白が挿入されることになります。

\$\$ agenda \leftarrow office \$\$
\$\$ \mathit{agenda} \leftarrow \mathit{office} \$\$

$agenda \leftarrow office$
 $agenda \leftarrow office$

上記の二つの例はいずれもアルゴリズムです。しかし、二つ目は正しい意味なのですが、一つ目は間違った意味になっています。執筆者の意図としては「リスト *agenda* に *office* を代入する」ということになりますが、一つ目は `\mathit` というコマンドを使っていないために、「変数 a, g, e, n, d, a の積に変数 o, f, f, i, c, e の積を代入する」という全く異なった意味になってしまいます。

このように文章表現を行う上では作文（組版）に関する約束事、知識を知らなければ読者に正確な意図が伝わらなくなります。

他とのコミュニケーションにおいて文字による伝達を採用する場合、それらに用いる記号の意味を正確に把握しなければ、「間違った意味」が相手に伝わることになります。文書の正確性が保持されていなければ、読者の深い理解と共感を得ることが難しくなります。

この冊子でもそのような「記号の使い方」に関する部分を取り扱い、それらを \LaTeX 上でどのように実現すればよいのかも説明します。このような文章表現に関する部分は

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ を用いない場合においても重要であると考えますので、本文中で強調して表記しています。

近年はワープロソフトと呼ばれるソフトウェアが多数存在します。OpenOffice.org とか Microsoft Office などがその類です。これらのワープロソフトと $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ とには決定的な差があります。ワープロソフトは文書の要素に直接視覚的な調整を施します。例えば、「T」という文字をワープロソフトで斜体にすると、強調を意味するのか変数を意味するのかという部分が曖昧になります。 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ をうまく使いこなせば、このような問題は発生しません。なぜならば、各要素に意味付けをすることができるからです。

1.4 $\text{T}_{\text{E}}\text{X}$ とは何か

$\text{T}_{\text{E}}\text{X}$ [12] とは Donald Knuth 氏によって開発された組版プログラムです。特筆すべきことは数式の処理に優れていること、簡単なレポートの作成から論文の作成、果ては商業出版にも耐えうる機能を持っていることなどです。

$\text{T}_{\text{E}}\text{X}$ の読み方ですがこれは英語ではなくギリシャ語の τ と ϵ と χ の綴りですから日本でこれに該当する発音がないと思われます。一般的には「てっく」と発音するのが無難だと思います。

1.5 WYSIWYG とは何か

WYSIWYG とは “What You See Is What You Get” の略で、「見たままのものが得られる」という意味合いでワープロソフトのように画面で見たイメージがそのまま紙などに出力されることを言います。

$\text{T}_{\text{E}}\text{X}$ は WYSIWYG ではありませんから紙に出力されるイメージをどうにかして確認する作業が必要になります。毎回紙に印刷するのは大変時間を必要とし、なおかつ地球環境の悪化を促進するものです。そのためコンピュータの画面上で確認作業をします。これをプレビューと言います。

1.6 一括処理とは何か

$\text{T}_{\text{E}}\text{X}$ のもう一つの特徴として通常のプログラミング言語と同じように原稿を一括で処理する方式を採用しています。これは当然のことなのですがワープロソフトとは大違いです。一括処理（バッチ処理）を採用しているということは、仕上がりは全てのページの組版が終了するまで分からないということです。マークアップ方式の言語ならば文書の全体をフォーマット（マークアップ付け）しなければならないのです。

1.7 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ とは何か

組版プログラムとしての $\text{T}_{\text{E}}\text{X}$ は完成度が非常に高く、高性能です。そのためちょっとした記事を書こうと思って手続きが多に多いようです。そこであらかじめいくつかの

命令を定義しておき、その定義を使って特定の書式を用意しておけば簡単に文書を作成することができます。このシステムを開発されたのが Leslie Lamport 氏で、彼の作成したシステムを \LaTeX [13] と言います。

\LaTeX も HTML と同様のマークアップ方式を採用しています。簡単な例を挙げると、

```
<CENTER>
  人類普遍の原理である
</CENTER>
```

という記述があるとします。これは「人類普遍の原理である」という文字列を中央に寄せたいので、「始まり」と「終わり」をそれぞれ、`<CENTER>` と `</CENTER>` という二つの規則で囲んでいます。これがマークアップ方式の典型的な例です。マークアップ方式ではそれぞれの要素に属性を与えて文書を記述するということを行います。HTML での表記が \LaTeX では、

```
\begin{center}
  人類普遍の原理である
\end{center}
```

となるので、先程の HTML の記述に良く似ているのが、お分かりになるでしょう。

\TeX も \LaTeX も欧文言語圏のためのプログラムですから標準では日本語を処理することが出来ませんが、中野賢氏を始めとするアスキーの方々が \TeX の日本語化をしてくださいましたので、今ではこの $\text{\TeX}/\text{\LaTeX}$ を使って高品質な日本語組版ができるようになりました。アスキーによって日本語化された \TeX や \LaTeX をそれぞれ \pTeX 、 \pLaTeX と呼びます。

1.8 情報の入手先

Leslie Lamport 氏の『文書処理システム $\text{\LaTeX} 2_{\epsilon}$ 』[13] やコンパニオンシリーズ [3, 4, 2]、入門用として奥村晴彦氏の『 $\text{\LaTeX} 2_{\epsilon}$ 美文書作成入門』[21]、それに藤田眞作氏の書籍 [9] や、乙部巖己氏と江口庄英氏による *Another Manual* シリーズ [23, 22, 1] が参考になると思います。以上の書籍は入手が容易だと思います。

\LaTeX やその周辺の情報を入手するにはインターネットを使って収集するのが良いでしょう。インターネット上で \LaTeX に関する情報交換がされています。

第 2 章

L^AT_EX の基本

まずは操作方法などの L^AT_EX の基本を説明します．コンピュータの基本操作に関する部分は大雑把にしか解説していませんので，適宜参考書を参照してください．

2.1 基本の基本

▼ 2.1.1 処理の流れ

コマンドを覚える前にまずは L^AT_EX での処理の流れをご覧ください．テキストファイルに文章そのものとコマンドというものを書き，それを L^AT_EX 処理し，成形結果を確認するといったことを何度か繰り返して最終的な版を仕上げます（図 2.1）．ここで成形とあり

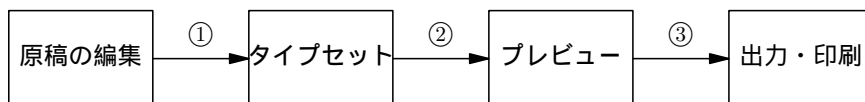


図 2.1 処理の流れ

ますが，L^AT_EX では元のソースファイルそのものに変更を加えて整形するのではなく，そこから新規に DVI ファイルというものを成形するのです^{*1}．

1. 原稿（ソースファイル）の編集

L^AT_EX を使うためには文章だけではなく，文章の構造や書式を決定するコマンドと呼ばれるものを記述します．この原稿をソースファイルと呼ぶこともあります．原稿はメモ帳や Emacs などのテキストエディッタで編集します．Unix 系 OS では

```
$ emacs file.tex &
```

とすると Emacs が立ち上がると思います．XEmacs でも何でも可能です．これが図 2.1 の①の矢印に対応します．

2. タイプセット（組版）

ソースファイルができたならそれを成形します．そのときに使うプログラムは欧文の

^{*1} 最近では L^AT_EX ファイルから直接 PDF を作るプログラム pdf L^AT_EX など存在しますが，日本語化されていません．

みの場合は latex , 日本語を扱うときは latex を日本語化した platex です . シェルやコマンドプロンプトなどのターミナルから ,

```
$ platex file.tex
```

とすれば文書が成形されます . この作業のことをタイプセットするとかコンパイルすると言います . これが図 2.1 の②の矢印に対応します .

3. プレビュー (確認作業)

今度はコンピュータの画面上で成形された結果を見ます . このとき $\langle file \rangle$.tex そのものが整形されるわけではなく新たに $\langle file \rangle$.dvi というファイルが作られます . これが L^AT_EX による組版後の文書になります . この組版後の結果をコンピュータ上で確認する作業のことをプレビューと言います . Unix 系 OS ならば

```
$ xdvi file.dvi &
```

などとすると良いでしょう . 最後のアンド ‘&’ があるとプログラムがバックグラウンドで起動しますので , 便利です . dviout をインストールした Windows ならばダブルクリックするだけで見られるでしょう . これが図 2.1 の③の矢印に対応します .

このような流れがあることを確認して実際に動くかどうかを試してみましょう .

▼ 2.1.2 動かしてみる

インストールが済んでいれば L^AT_EX が動きます . インストールまで進んでいないという方は近くの詳しい方に聞いてみてください .

とりあえず自分のいつも使っているテキストエディッタ (メモ帳や Emacs など) で以下のようなファイル first.tex

```
\documentclass{jarticle}
\begin{document}
こんにちは\LaTeX !!
\end{document}
```

を作成してください . Unix 系 OS ならば Emacs で良いでしょう . 日本語を打ち込むためには Emacs の場合はまず Emacs のウィンドウ下部に注目してください . ウィンドウの下部の表示は

```
[~]J.:---Emacs: first.tex (LaTeX)-[L1-All----]
```

となっていると思います . 一番右側の [~] という部分が半角入力 (英数入力) か全角入力 (日本語入力) かの違いを表します . うえの状態は英数文字の入力ができます . ここで半角・全角の入力を切り替えるためには **CTRL** を押しながら **¥** を押します . すると [~] という表示から **[あ]** という表示になると思います . 表示は使っているかな漢字変換プログラムによって若干違うかもしれませんが ^{*2} . **[あ]** の状態ですと日本語が入力で

^{*2} **Shift**+**Space** で切り替える場合もあります .

きる状態です．最近のパソコンと呼ばれるコンピュータには[半角/全角]というキーがあり，Windows の場合はそれで半角と全角の切り替えを行います，Unix 系 OS は違いますので注意してください．詳しくはご自分のテキストエディタのマニュアルを見るなどの対応をしてください．

次はタイプセット作業（ \LaTeX 処理，またはコンパイル）をします．Windows ならば[スタート]メニューから[ファイル名を指定して実行]というメニューがあるので，そこに `command` と入力して ‘OK’ ボタンを押せばコマンドプロンプトが起動するはずで，そしてシェル上やコマンドプロンプトでファイルが存在するディレクトリ（Windows の方はフォルダ）に移動して

```
$ platex file.tex
```

としてタイプセットしてください．すると端末（ターミナル）に

```
「 This is pTeX, Version 3.14159-p3.0.4 (sjis) (Web2C 7.3.9)
  pLaTeX2e (based on LaTeX2e <2001/06/01> patch level 0)
  (1) (./file.aux)
  Output written on file.dvi (1 page, 236 bytes).
  Transcript written on file.log.
」
```

のように表示されると思います．始めにバージョン情報を表示して終わりには $\langle file \rangle$.dvi に組版後のファイルを出力し，処理状況を $\langle file \rangle$.log に書き出したことになっています． $\langle file \rangle$.tex をタイプセットして出力された $\langle hoge \rangle$.log にはエラーメッセージなどの重要な情報が書かれているときがあるので何か問題が発生したときは眺めてみると良いでしょう．

タイプセット後にはいくつかのファイルが生成されています．ls (Windows の方は dir) コマンドで

```
$ ls file.*
```

とすると

```
「 file.aux file.dvi file.log file.tex
」
```

の四つのファイルが存在することを確認してください． \LaTeX の原稿であるソースファイル $\langle file \rangle$.tex をタイプセットしただけで三つもファイルが生成されたので，これらのファイルが何であることを簡単に説明します．

$\langle file \rangle$.aux 次回のタイプセットに必要な中途ファイル．目次の作成や相互参照をするために必要なファイル．

$\langle file \rangle$.dvi $\langle file \rangle$.tex をタイプセットして出来上がった印刷できる成形ファイル．DVI ファイルと呼ばれる．

$\langle file \rangle$.log $\langle file \rangle$.tex をタイプセットしたときの処理状況やどのような流れで処理をしたのかが書いてあるログファイル．

$\langle file \rangle$.tex 先程作成した \LaTeX の原稿であるソースファイル．

▼ 2.1.3 原稿作成時の注意点

これまでの作業ができていれば数式や図表を含まない簡単な文書を作成できることでしょう．そして実際に長い文章を打ち込んでみてください．ただし 10 個の半角記号

\$ % & _ { } ~ ^ \

は特殊文字として L^AT_EX に別の仕事をさせるために使いますので，そのまま使うことができません．さらに 3 個の記号は出力が違う文字記号になります．

| < > はそれぞれ — i j

となることでしょう．以上の 13 個の記号を文章中で出力するために面倒ですが，バックスラッシュ (\) 記号を補ったり長い命令を打ち込みます．

\# \\$ \% \& _ \{ \}

\$ % & _ { }

\textasciitilde \textasciicircum

\textbackslash \textbar

\textless \textgreater

~ ^ \ | < >

▼ 2.1.4 フォルダ・ファイルの基本的な操作

ターミナル上でのディレクトリの移動方法を知らない，フォルダの作り方を知らないという方のために，コマンドプロンプトやシェルでの主要なコマンドを紹介します．まず Windows では表 2.1 などの基本的なコマンドが提供されています．

表 2.1 Windows OS の基本コマンド

コマンド名	意味
mkdir	新規にフォルダ作成
cd	フォルダの移動
dir	ファイルの情報を表示
move	ファイル名を変更・移動
copy	ファイルをコピー
del	ファイルを削除
help	コマンドのヘルプ表示

表 2.2 Unix 系 OS の基本コマンド

コマンド名	意味
mkdir	ディレクトリの作成
cd	ディレクトリ移動
ls	ファイルの情報表示
mv	ファイル名の変更・移動
cp	ファイルをコピー
rm	ファイルを削除
help	内部コマンドのヘルプ表示
man	コマンドのヘルプ表示

それぞれのコマンドの使い方 (ヘルプ) は `command /?` とすることで表示できます．または [スタート] メニューの [ヘルプ] からコマンドプロンプトについて調べてみても同様のことができます．どちらかというと Windows ヘルプを利用したほうが良いでしょう．コ

マンドプロンプトなどの操作に慣れていないという方は 9.9 節を参照して \LaTeX の入力支援環境を使うのも良い方法です。

Unix 系 OS の方はコマンドを覚えなければ操作に不便を感じると思われますので、日ごろから使うように習慣付けをすると良いでしょう。シェルと言っても何種類がありますし、シェルに関しては 1 冊の本になるくらい奥の深いものなので詳細はそれらに譲ります。ここでは基本的なファイル操作のコマンドだけを表 2.2 に紹介します。それぞれのコマンドの簡単なヘルプが見たいときは `command --help | less` のようにすると `less` がページを整形します。もう少し詳しいヘルプが見たいときは `man command` とします。もっと詳しいヘルプが見たいときは `info command` のようにすると `info` がページを整形します。less や info の操作方法は若干癖がありますので慣れるまで時間がかかるかもしれません。Unix 系 OS ならば

```
$ emacs file.tex &
$ platex file.tex
$ xdvi file.dvi &
```

の三つの操作ができればなんとかなります。この冊子では Unix 系 OS の基本ツールなどまで詳しく解説しないのでご自分で調べてみてください。

コマンドに対してシェル上で一緒に渡す文字列のことを引数と呼びます。そして多くのコマンドはコマンドラインオプションといってハイフン ‘-’ がハイフンが二つ ‘--’ で始まる引数を特別なスイッチとして扱います。このスイッチによってそのコマンドは挙動を変えます。それぞれのコマンドでどのようなコマンドラインオプションが使えるのかは各プログラムのヘルプを調べます。

▶ 問題 2.1 以下の作業をターミナル上から行ってください。Windows の方は `mv` を `move` に、`ls` を `dir` に、スラッシュ ‘/’ を円 ‘¥’ と置き換えてください。

```
$ echo message1 >> file.txt
$ echo message2 >> file.txt
$ mkdir anydir
$ cd anydir
$ mv ../hoge.txt ./
$ ls
$ more file.txt
$ ls ../
```

上記の操作はどのような結果をもたらしたと考えられるでしょうか。

新規にディレクトリ `anydir` を作成し、現在のディレクトリ（カレントディレクトリ）に存在していたファイル `file.txt` を `anydir` ディレクトリに移動したと考えられるでしょう。最後の操作でうえの階層のディレクトリ（親ディレクトリ）に `file.txt` がないことでそれを確認できます。

▼ 2.1.5 エラーに遭遇する

L^AT_EX 処理をしているとエラーに悩まされるかもしれません．L^AT_EX は文章中にコマンドなどに関するエラーを発見するとそこで処理を中断します．処理を中断するとユーザにどうすれば良いかを促します．このときターミナルには疑問符‘?’が表示されます．

▷ 例題 2.2 まずは以下のファイル `error.tex` を作成してください．

```
\documentclass{jarticle}
\begin{document}
Hello & Goodbye! Give me $100! Give me 100%!
Under_bar is stranger. Is sharp sing #?
No its' \#. Hello \& Goodbye!!
\end{document}
```

次に `error.tex` を `platex error` でタイプセットしてください．するとターミナルには

```
「 ! Misplaced alignment tab character &.
1.3 Hello &
      Goodbye! Give me $100! Give me 100%!
?
」
```

と表示されるでしょう．最後の行に疑問符‘?’が表示されています．この状態はユーザに何らかの操作を促している状態です．どうやら3行目でアンド‘&’を不正に使っていると言われています．ここで `Enter` キーを押すとさらに

```
「 ! You can't use 'macro parameter character #' in math mode.
1.4 Under_bar is stranger. Is sharp sing #
?
」
```

と表示されます．シャープ‘#’も間違った使い方をしていると指摘されました．さらにもう1度 `Enter` キーを押すと

```
「 ! Missing $ inserted.
<inserted text>
      $
1.6 \end{document}
?
」
```

という表示になります．今度はドル‘\$’を不正に使ったと言われました．以上のことから3行目から6行目にかけて半角記号の使い方が間違っていることが分かりました．ソースファイルをもう1度確認し，どこがどう違うのかを判別し修正してください．

修正後のファイルは以下になるでしょう．

```
Hello \& Goodbye! Give me \$100! Give me 100%!
Under_bar is stranger. Is sharp sing \#?
No its' \#. Hello \& Goodbye!!
```

3行目のアンド‘&’とドル‘\$’と4行目のナンバー‘#’にバックスラッシュ‘\’を付けます．これを再びタイプセットしてみてください．今度は

```

「 ! Missing $ inserted.
  <inserted text> $
  1.4 Under_
      bar is stranger. Is sharp sing \#?
  ?
」

```

ドル '\$' の書き忘れがあるとされています。4 行目のエラーメッセージで丁度アンダーバー '_' の部分で表示が改行されていますから、この部分に間違いがあることが分かります。どうやらアンダーバーはドル '\$' などと同じようにバックスラッシュが必要なようです。ここでとりあえず **Enter** キーを押してタイプセットを終了してください。一つ目のエラーとして

```

「 ! Missing $ inserted.
  <inserted text> $
  1.6 \end{document}
  ?
」

```

が表示されます。タイプセットは中断しませんが

```

「 Overfull \hbox (152.35132pt too wide) in paragraph at lines 3--6
  []\OT1/cmr/m/n/10 Hello & Good-bye! Give me $100! Give me
  100Under$[]\OML/cmm/m/it/10 arisstranger:Issharpsing
  \OT1/cmr/m/n/10 \#?\OML/cmm/m/it/10 Noits[]\OT1/cmr/m/n/10 #
  \OML/cmm/m/it/10 :Hello\OT1/cmr/m/n/10 &\OML/cmm/m/it/10
  Goodbye\OT1/cmr/m/n/10 !!$
」

```

という煩雑な表示が出現します。これは **Overfull \hbox** という警告であることが分かります。次に成形後の DVI ファイル error.dvi をプレビューしてください。すると行がページをはみ出しています。先程のアンダーバーに関するエラーにおいて

```

「 <inserted text> $
」

```

という表示がありました。どうやら L^AT_EX は自動的にドル '\$' を挿入したようです。'b' という文字が 'r' の下付きの添え字になっています。さらに 'Give me 100Under_b' となっており入力されたパーセント '%' と感嘆符 '!' が出力されておらず、次の行の 'Under' とくっついています。どうやら先程の煩雑な警告はこの行がページをはみ出していることを意味しているようです。ですからファイル error.tex はさらに次のように修正することになります。

```

Hello \& Goodbye! Give me \$100! Give me 100\%!
Under\_bar is stranger. Is sharp sing \#?
No its' \#. Hello \& Goodbye!!

```

これで望み通りうまく行きそうです。実際に上記のファイルをタイプセットし、その結果を吟味してください。

L^AT_EX の原稿をタイプセットしたときに端末に疑問符 '?' が表示されて処理が中断しますが、この段階でこちらも疑問符 '?' で返事を返すと

```

「 Type <return> to proceed, S to scroll future error messages,
」

```

```

R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
?

```

と表示されます。疑問符「?」が表示されている段階で上記に挙げるようなキーの入力をする
と何らかの対処ができることになっています。

- [Enter] エラーに対して L^AT_EX が適当な対処をした後にタイプセットを続行します。
- [S] [Enter] キーを押し続けたことと同じ動作をします。
- [R] エラーが検出されても停止せずにノンストップでタイプセットします。
- [Q] [Q]を押した場合はバッチモードに入り処理が続きます。
- [I] 〈文字列〉 文字列を挿入してタイプセットを続けます。元の原稿に修正は加えられません。
- [H] そのエラーに対する英語のヘルプをターミナルに表示します。
- [X] タイプセットを終了します。

[X] キーは余り押しはいけません。括弧が足りないというエラーの場合はとりあえず [Enter] キーを押せばそのまま処理を続行できます。

タイプセットをしてアスタリスク「*」が表示されて処理が中断するときがあります。

[Enter] キーを押しても同じメッセージが表示されてどうにもなりません。

```

*
(Please type a command or say '\end')

```

この場合コンソール上から

```
$ \end{verbatim*}
```

と入力して処理が終了しなかった場合は強制的にプログラムを終了してください。ソース
中で何かミスをしていると思われます。

▼ 2.1.6 プレビューの操作

プレビューを行うプログラムのことをプレビューアと言います。OS によって使用可能な
プレビューアが異なります。Windows ならば大島利雄氏の dviout, Unix 系 OS なら
ば xdvi, Red Hat ならば pxdvi などを使い *(file).dvi* を各アプリケーションで開きます。
Windows の場合は dviout に関する豊富なヘルプやマニュアルが用意されているのでそ
ちらを読んでみてください。ここでは Unix 系 OS で広く使われている xdvi を例に操作
方法を説明します。まずシェル上で hoge.dvi の存在するディレクトリに移動し、xdvi
に対してファイル名を指定し、

```
$ xdvi hoge&
```


のようにします。Unix 系 OS ならばアンド '&' をつけてバックグラウンドで起動します (図 2.2)。こうするとタイプセットを再度したときに自動的に DVI ファイルを再表示します。dviout でも同様の再表示機能があります。

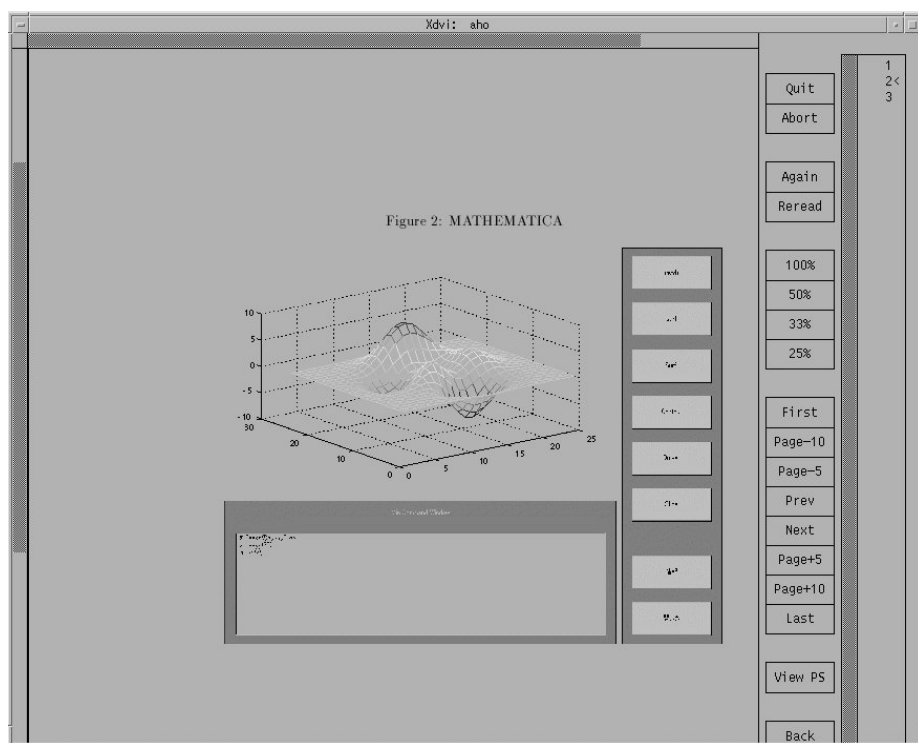


図 2.2 xdvi の起動例

xdvi の基本的な操作方法を説明します。右側に枠で囲まれた文字がボタンになっています。ボタンのように見えませんが一応押せます。さらにボタンの右側にはページ番号があり、ページ番号をクリックすると該当するページを表示します。

xdvi でのマウスのクリックは拡大の機能を持っています。それぞれ

- 左クリック 少し拡大。
- 中央クリック 普通に拡大。
- 右クリック かなり拡大。

となっています。また、右側にある 'Quit' とか 'Abort' などはボタンで、主なボタンの機能は以下のとおりとです。

- Quit xdvi を終了する。
- Reread 一度読み込んだファイル *<file>*.dvi を再描画する。
- First 先頭ページに移動する。
- Prev 前ページに移動する。
- Next 次ページに移動する。

Last 最終ページに移動する .
 View PS PostScript ファイルを見る .
 File DVI ファイルを別に開く .

終了するには ‘Quit’ ボタンを押します .

▼ 2.1.7 コマンド

L^AT_EX では原稿を三つのパートに分割することができます . それに伴い , いくつかのコマンドは , 特定のパートでしか使用できません .

原稿先頭部分 (イニシャルコマンド) <code>\documentclass[〈オプション,...〉]{〈クラス〉}[〈リリース〉]</code> 前書き部分 (プリアンブルコマンド) <code>\begin{document}</code> 本文 (ボディ) <code>\end{document}</code>

この中で `\documentclass` 命令と `document` 環境は必須であり , 絶対に必要な記述です .

原稿先頭 (イニシャル) 部分にはイニシャルコマンドと呼ばれるコマンドを記述することができ , 同じように前書き (プリアンブル) 部分にはプリアンブルコマンドや定義などを記述することができます . そして , `document` 環境によって挟まれた本文部分にはコマンドの定義や組版用のコマンドを記述します . それぞれのコマンドは定められた場所で使うように決められています . ユーザがプリアンブルコマンドを本文で使うことができないように L^AT_EX の内部で細工が施されています .

ここで言葉の定義をしておきます . コマンド , 命令 , 環境 , 引数 , オプションなどの言葉を混同しがちですが , この冊子では以下のように取り決めます .

コマンド バックスラッシュ (Windows の方は円記号) と共に用いられる文字列 .

命令 単独で使用するコマンド . 引数を取ることができる .

例 : `\alpha` , `\maketitle`

環境 ‘`\begin{〈何々〉}`’ と ‘`\end{〈何々〉}`’ で囲まれている領域 , またはそれを囲むためのコマンド . 引数を取ることができる .

例 : `\begin{center}`文字列`\end{center}`

引数 コマンドに受け渡す文字列 .

必須引数 波括弧 ‘`{ }`’ で囲まれた要素 . コマンドが必須引数を取るときは必ず受け渡す .

例 : `\section{引数}`

任意引数 オプションとも言う . 角括弧 ‘`[]`’ で囲まれた要素 . コマンドが任意引数を取るときは任意に受け渡す .

例 : `\documentclass[任意引数]{jbook}`

▼ 2.1.8 括弧について

さて、L^AT_EX の基本を知ったところで括弧についての取決めをしたいと思います。括弧については色々な呼び方があるようですが、誤解を避けるためにこの冊子では以下のように定義します。

かぎ括弧—「」 引用や会話文などに使う。

二重かぎ括弧—『』 書名、引用の中の引用などに使う。

引用符—‘ ’ シングルクオートとも言う。左側にあるほうを左シングルクオート、右側にあるほうを右シングルクオートと言う。引用に使う。

二重引用符—“ ” ダブルクオートとも言う。左側にあるほうを左ダブルクオート、右側にあるほうを右ダブルクオートという。長い引用に使う。

丸括弧—（ ） 小括弧、パーレンとも言う。語句の補足説明に使う。

波括弧—{ } 中括弧とも言う。コマンドに対して必須引数を渡すのに使われたり、要素を一つのグループにまとめるために使う。

角括弧—[] 大括弧とも言う。コマンドに対して任意引数を渡すときに使う。

山括弧—< > この括弧に囲まれた文字列は何か別の文字列に書き換えられる。例えば、〈ファイル名〉などがあれば、これは任意の文字列 `file.tex`, `input.foo`, `output.bar` などに置き換えられる。

ここで引用符と言うのが登場しましたが、欧文の引用符はシングルクオート（‘ ’）であり、和文の引用符はかぎ括弧（「」）となります。二つを区別するために欧文用のものをシングルクオート、和文のものをかぎ括弧と言うことにします。文中に出てくる引用符という言葉はそのどちらも示すことになります。

2.2 L^AT_EX に関わるファイル形式

タイプセット時に作成される中途ファイル以外にも L^AT_EX では多くのファイル形式が存在するを経験するでしょう。一般にファイル形式は拡張子によって種類を識別します。

〈ファイル名〉. 拡張子

のようにピリオドの後の文字で区別されます。

パッケージをインストールするときに見かけるものは以下の通りです。

.dtx パッケージ化されたマクロ。複数のクラス〈クラス 1〉.cls, 〈クラス 2〉.cls, ... 〈クラス n〉.cls が〈クラス〉.dtx 中にまとまっていることも多い。または〈マクロ〉.sty が複数まとまっているときもある。

.ins パッケージ化されたマクロを取り出すためのファイル。〈classes〉.dtx とともに配布されている。

.sty 便利な機能をうまくまとめたもの。マクロ、マクロパッケージ、パッケージ、ス

タイルファイルとも言う．

- .cls 原稿の書式を決定するファイル．クラス，クラスファイル，文書クラスファイル，ドキュメントクラスファイルとも言う．
- .clo クラスのオプションに応じた設定を記述したファイル．
- .fd 書体の属性を定義したファイル．ユーザが意識して使うことはない．

原稿を作成するときに見かけるものは以下の通りです．

- .tex L^AT_EX が処理を受け付ける原稿．ソース，ソースファイルとも言う．
- .bib 文献成形プログラム Bib_TE_X が処理できる参考文献ファイル．参考文献データベースと言う．
- .bst 参考文献の表示形式を決めるもの．参考文献スタイルと言う．
- .eps Adobe 社が開発したページ記述言語 PostScript で書かれたファイル．主にベクトル画像などに使われる．

原稿をタイプセットした後に見かけるものは以下の通りです．これらは全て中途ファイルであり，L^AT_EX が原稿を完成させるために必要なものです．

- .log L^AT_EX の組版結果の詳細情報．ログファイルと言う．
- .aux 相互参照などの情報が書かれたファイル．1 度目以降の処理に必要とされる．
- .dvi 原稿を L^AT_EX でタイプセットした後に作成される印刷結果に限りなく近いファイル．このファイルをプレビューしたり，または他のデバイスドライバによって別の形式に変換できる．
- .toc 「目次」を出力するための目次情報が書き出されたファイル．
- .lof 「図目次」を出力するための図目次情報が書き出されたファイル．
- .lot 「表目次」を出力するための表目次情報が書き出されたファイル．
- .bb1 Bib_TE_X によって並べ替えをした後の参考文献リスト．thebibliography 環境を用いて記述されている．
- .blg Bib_TE_X の実行結果が出力されるログファイル．

その他画像形式に関わる拡張子として主に以下があります．

- .jpg 写真などのフルカラーに適したビットマップ画像．
- .bmp Windows OS 標準の無圧縮ビットマップ画像．
- .png 可逆圧縮で Dvips 対応しているビットマップ画像．
- .bb L^AT_EX が画像のバウンディングボックス情報を得るために必要とするファイル．ebb や CrateBB で作成できる．
- .mp MetaPost で描画されたベクトル画像．

2.3 コマンドの基本

L^AT_EX では便利なコマンドがあらかじめ用意されています．それらをどのように用いる

か、また必要な機能がないときはどうすれば良いのかを説明します。

▼ 2.3.1 プリアンブルでのコマンド

原稿のプリアンブルに書くべきコマンドは `\documentclass` 命令です。

```
\documentclass[〈オプション,...〉]{〈クラス名〉}[〈リリース〉]
```

この命令は体裁を決定する書式ファイルを決めるという意味合いを持っており、この命令を書いた後は原稿の前書き部分（プリアンブル）として解釈されます。

〈クラス名〉には 3.21.1 節で紹介するものが使えます。〈オプション〉にはそれぞれのクラスが用意している任意引数を渡すことが出来ます。このオプションのことを特に文書クラスオプションとかドキュメントクラスオプションと言います。〈リリース〉には自分の使っているクラスファイルがいつ配布されたのかを書きます。

〈リリース〉にはクラスの配布された日付を 〈YYYY/MM/DD〉 という書式で記述できます。例えば、2003 年 12 月 31 日に公開された日本語のクラス `jarticle` ならばおおむね以下ようになります。

```
\documentclass[11pt,a4j]{jarticle}[2003/12/31]
```

もしも、クラスファイルが 2003 年 12 月 31 日以前のもので要求されているバージョンよりも古ければ、 \LaTeX はタイプセット時に警告（Warning）を出します。

他にも 3.21.3 節で紹介しているようなパッケージを使う場合はプリアンブル部分に `\usepackage` を使います。

```
\usepackage[〈オプション,...〉]{〈パッケージ名〉}[〈リリース〉]
```

これはプリアンブルのみでしか使えません。`\usepackage` 命令は `\documentclass` 命令と同じように、そのパッケージが提供するオプションを指定したり、リリースにはそのパッケージのバージョンを指定できます。例えば、画像ファイルなどを \LaTeX で扱いたいと思い、デバイスドライバとして `Dvipdfm` を使う場合は

```
\usepackage[dvipdfm]{graphicx}[2001/01/01]
```

のように `graphicx` パッケージを使うことをプリアンブルで宣言します。

同じパッケージを 2 度や 3 度以上読み込もうとしても、1 度読み込まれているなら再度読み込もうとしません。パッケージに渡すオプション（リリースを除く）を特にパッケージオプションと呼びます。

文書クラスオプションやパッケージオプションのいずれにしても、たいてい「命令」と「必須引数」のあいだの 〈オプション〉（任意引数）は複数個渡すことができます。例えば

```
\documentclass[10pt,a4paper,twocolumn]{article}
```

のように `10pt`、`a4paper`、`twocolumn` という三つのオプションはコンマ‘,’を区切りとして書けば良いのです。

同時に複数のパッケージを使うことも宣言できます。graphicx, amsmath, makeidx などを

```
\usepackage{graphicx,amsmath,makeidx}
```

のように宣言できますがパッケージオプションをそれぞれのパッケージに対して渡すことはできません。

基本的なソースファイルは次のようになります。

```
\documentclass[10pt,a4paper,oneside]{jarticle}
\usepackage[dvipdfm]{graphicx}
\usepackage[dvipdfm,usenames]{color}
\begin{document}
ここに本文を書きます。
\end{document}
```

後述のデバイスドライバの指定に関しては、上記のような記述ではなく、ドキュメントクラスオプションに使用するデバイスドライバを追加するのが安全です。

```
\documentclass[dvipdfm,10pt,a4j]{jarticle}
\usepackage{graphicx}
\usepackage[usenames]{color}
```

第 3 章

文章の書き方

L^AT_EX で文書を作成するためには文章の組版に関する約束事を知る必要があります。論理的な文章を書きたいと思ったら、その論理を知る必要があります。この章ではそれらを L^AT_EX で実現するための基本的な部分を説明します。

3.1 文章の論理構造

一般的な文書 (document) を作成するうえで覚えたほうが良い項目を示します。

表 題 (title) 文書には必ず表題をつけて誰 (`\author`) がいつ (`\date`)、何 (`\title`) を作成したのかを示します。

目 次 (contents) ページが多い場合には目次をつけて読者に参照しやすいようにします。大規模な文書の場合、読者はまず目次を参照し、その文書を読むべきかどうかを判断しますので、学位論文などでは目次は必須項目です。

見出し (headline) 見出しを付けてこれから何について話をするのかを明確にします。見出しは目次と関連していますので、読者がすぐに理解できるようにします。

段 落 (paragraph) 一つの話題について一区切り付いたら段落を分けます。

字下げ (indentation) 段落始めは全角 1 文字ほど開けて字下げを行ないます。欧文の場合、見出し直後の字下げは慣習的に行ないません。

句読点 (punctuation) 文章の中で文の区切り、文の終わりには句読点などの区切り記号を付けます。

注 釈 (note) 難解と思われる用語、補足すべき情報があれば注釈として添えます。注釈はあくまで補足情報であって、読者がその注釈を読まなくても、何ら影響がないようにします。

このような構造は日本語や他の言語でもほとんど共通です。誰かに何かを文書で伝えるときにはこのような構造が必要になります。文書の最小構成単位は単語 (word) です。文字 (character) から文 (sentence) ができ、段落 (paragraph) ができ、節 (section) ができ、章 (chapter)、部 (part) へとつながっていきます。日本語の場合は表意文字なので最小単位は文字 (letter) に相当します。

表 3.1 文書の構成要素

文字	単語	文	段落	節	章	部
letter	word	sentence	paragraph	section	chapter	part

\LaTeX はユーザが約束通りにコマンドを打ち込み文章を練り上げていけば、字下げ、相互参照、図表の配置、目次の作成など、様々なことを半自動的に行ってくれます。ここではその基本的な約束を紹介します。

3.2 表題

表題はその文書が何について書かれたものなのかを示すために必要な要素です。通常は題名 (title), 作者 (author), 日付 (date) を書くのが一般的ですからプリアンブルに

```
\title{\<題名>}
\author{\<作者>}
\date{\<日付>}
```

の三つを書き込みます。 \LaTeX ではプリアンブルに表題の情報を書き込んでも出力まではしませんので `\begin{document}` の後に

```
\maketitle
```

とします。

例を示すと入力が

```
\documentclass{jarticle}
\title{はじめての\LaTeX}
\author{未来 太郎}
\date{2004 年 3 月 30 日}
\begin{document}
\maketitle
{\LaTeX}を使うのはこれが初めてです。
\end{document}
```

であったならば、大体の出力は以下ようになります。

はじめての \LaTeX

未来 太郎

2004 年 3 月 30 日

\LaTeX を使うのは...

3.3 見出し

文書に見出し (sectioning) と目次 (contents) がなければ、記事の検索に時間がかかるのは容易に想像できるでしょう。そこで、文書の中には階層的な見出し (nested sections) を作成します。またその文書の概略が存在すればその文書に何が書かれているのかがすぐに分かるので、概要 (abstract) を付け足すのも効果的です。

▼ 3.3.1 見出しの出力

文書の中の一連の段落に何が書かれているのかを分かりやすくするために見出しを記述します。また見出しは同一ページに同じ名前のものが存在しても良いように通し番号をつけて一意的に管理します。

L^AT_EX での見出しの定義は表 3.2 の通りです。\\section などの見出し命令を使って見

表 3.2 L^AT_EX での見出しの定義

\\part[<目次用の見出し>]{<見出し>}	部
\\chapter[<目次用の見出し>]{<見出し>}	章 *
\\section[<目次用の見出し>]{<見出し>}	節
\\subsection[<目次用の見出し>]{<見出し>}	項 (小節)
\\subsubsection[<目次用の見出し>]{<見出し>}	目 (小小節)
\\paragraph[<目次用の見出し>]{<見出し>}	段落
\\subparagraph[<目次用の見出し>]{<見出し>}	小段落

*article や jarticle では定義されていません。

出しを作成します。前後の空白の調節や改ページ、改行、書体の変更などはほぼ自動的に行われ、通し番号 (serial number) が付加されます。‘[<目次用の見出し>]’ という任意引数がありますが、これは見出しが非常に長いときに、それを短縮した文字列を目次に書き出すようにします。別に長いときだけでなく、見出しと目次の文字列を別にしたいときなどにも使えるでしょう。使い方は簡単です。見出しを階層構造的に書き記せば、L^AT_EX は自動で階層ごとに番号付けをします。例としては次のような通し番号が振られます。

\\chapter{特殊相対性理論}	第 1 章 特殊相対性理論
\\section{歴史的背景}	1.1 歴史的背景
\\chapter{一般相対性理論}	第 2 章 一般相対性理論
\\section{電気学との関連}	2.1 電気学との関連
\\subsection{電気の次元数}	2.1.1 電気の次元数

▼ 3.3.2 見出しの深さ

L^AT_EX ではあらかじめ部 (part)、章 (chapter)、節 (section)、小節 (subsection)、小小節 (subsubsection)、段落 (paragraph)、小段落 (subparagraph) という七つの見出し用

表 3.3 見出しの階層

見出し	命令	深さ *
部	<code>\part</code>	-1 (0)
章	<code>\chapter</code>	0 (なし)
節	<code>\section</code>	1
小節	<code>\subsection</code>	2
小小節	<code>\subsubsection</code>	3
段落	<code>\paragraph</code>	4
小段落	<code>\subparagraph</code>	4

* 括弧内は (j)article での深さ

文章の論理構造を整理するとき、一つの文書を項目ごとに分けることができます。さらにその項目を小項目で分けることもできるわけです。小項目があると文書の構造は階層的になります。項目が分かれていることを区別するために見出しを付けます。見出しを目次としてひとまとめに出力すると、読者は目的の項目を探しやすくなります。

のコマンドを用意しています。ただし (j)article などでは章は用意されていませんし、クラスによって深さが若干違います。

3.4 目次の出力

目次は見出しから読みたい箇所に移動するための見出し一覧です。これは 20 ページ以上の文書にあることが望まれます。目次といっても \LaTeX では

```
\tableofcontents (目次 (contents) を出力するための命令)
\listoffigures (図目次 (List of Figures) を出力するための命令)
\listoftables (表目次 (List of Tables) を出力するための命令)
```

の三つの命令が用意されており、それぞれ出力したい場所に命令を書きます。注意すべきこととして、目次を作成するためには最低 2 回のタイプセットをします。

▼ 3.4.1 目次を出力する深さ

目次をどの階層まで出力するかはカウンタ `tocdepth` の値を表 3.3 に従って変更します。jsbook などでは章 (`\chapter`) まで出力したいならば

```
\setcounter{tocdepth}{0}
```

のようにします。(j)book と (j)report の標準は 2、(j)article ならば 3 です。jsbook は 1 になっています。

▼ 3.4.2 見出しの番号付けの深さ

見出しの通し番号はカウンタ `secnumdepth` によってどの階層まで出力するかを決められます。`secnumdepth` の値は表 3.3 に従って変更します。少節 (`\subsection`) までに番号を付けるようにするには

```
\setcounter{secnumdepth}{2}
```

のようにします。これは目次側にも影響します。

3.5 概要の出力

文書の概略が存在すればその文書に何が書かれているのかが大まかに分かるので概要 (abstract) を書くのが良いでしょう。「概要」は「はしがき」とも呼ばれ、文書クラスによって出力方法が違います。(j)article 系ならば abstract 環境を使います。この abstract 環境は \maketitle 命令と関わりがあるので概要を出力するためには \maketitle 命令の後に書きます。

```
\maketitle
\begin{abstract}
<文書の概要>
\end{abstract}
```

次に (j)report の場合ですが概要専用の環境は用意されていません。そこで概要を章立てすると良いので \chapter* 命令を使います。このとき \chapter 命令にアスタリスク '*' をつけると目次に見出しを書き出さず、章番号を付け足しません。使用例は

```
\chapter*{概要}\addcontentsline{toc}{chapter}{概要}
```

と記述してから概要の文章を書きます。標準の文書クラスでは概要専用のコマンドは定義されていません。

最後に (j)book の場合ですが、これは \frontmatter が宣言されているときに \chapter 命令を使うと余計な手間を省くことが出来ます。具体的には

```
\begin{document}
\frontmatter% 前付け
\chapter{まえがき}
ここに概要やまえがきを書きます。
\mainmatter% 本文
\chapter{序論}
```

とすると目次にも概要を番号なしで書き出します。

3.6 段落と字下げ

文章で段落をはじめようと思えば、まず字下げ (indentation) をします。この字下げの作業を L^AT_EX は半自動で行います。使い方は

天皇は、日本国の象徴であり日本国民統合の象徴であつて、
この地位は、主権の存する日本国民の総意に基く。

皇位は、世襲のものであつて、国会の議決した皇室典範の
定めるところにより、これを継承する。

天皇の国事に関するすべての行為には、内閣の助言と承認

を必要とし、内閣が、その責任を負ふ。

のように 1 行空けて入力すれば

天皇は、日本国の象徴であり日本国民統合の象徴であつて、この地位は、主権の存する日本国民の総意に基く。

皇位は、世襲のものであつて、国会の議決した皇室典範の定めるところにより、これを継承する。

天皇の国事に関するすべての行為には、内閣の助言と承認を必要とし、内閣が、その責任を負ふ。

として自動的に字下げがなされます^{*1}。明示的に `\par` 命令で段落の終了を知らせることができ、以下のようにも書けます。

天皇は、日本国の象徴であり日本国民統合の象徴であつて、
この地位は、主権の存する日本国民の総意に基く。`\par`
皇位は、世襲のものであつて、国会の議決した皇室典範の
定めるところにより、これを継承する。`\par`
天皇の国事に関するすべての行為には、内閣の助言と承認
を必要とし、内閣が、その責任を負ふ。`\par`

以上のように \LaTeX はワープロソフトとは違い、原稿中の一つの改行が出力と対応していないのがお分かりになるでしょう。 \LaTeX では改行すべき位置を自動で計算しているのです。

字下げの幅は `\parindent` という長さ変数で指定されているので

```
\parindent=3em
```

とすると約 3 文字分の字下げを段落の始めで行うことができます。

▼ 3.6.1 行頭の子下げの有無

段落の開始には字下げをすべきなのですが、何らかの理由により字下げを抑制したいときがあります。字下げの有無に関しては `\indent` と `\noindent` 命令が使えます。

```
\indent 可能ならば字下げをします。  
\noindent 可能ならば字下げをしません。
```

`jreport` などのクラスファイルではこのような命令を使っても行頭の字下げができないときがあります。その場合は `indentfirst` パッケージを読み込みます。

```
%\section{字下げ}
```

```
\indent 私は\indent 大学生ですから、そうなり      私は大学生ですから、そうなります。  
ます。 \par                                         そうなりました。
```

```
\noindent そうなりました。
```

^{*1} 『日本国憲法』 1947 年 5 月 3 日施行の第 1 条から第 3 条までの引用。

▼ 3.6.2 ダブルスペース

ダブルスペースといって行送りを倍にするということも迫られる場合があります．これは行送りの量を決める長さ変数 `\baselinestretch` を

```
\renewcommand{\baselinestretch}{2}
```

とする方法があります．他にも `doubleSPACE` パッケージを

```
\usepackage{doubleSPACE}
```

としてプリアンブルで読み込むと上記の方法よりもうまくダブルスペースを実現できます．

3.7 長さの単位

▼ 3.7.1 L^AT_EX での単位の取り決め

先ほどは何らかの変数（パラメータ）に数値を代入する

```
\parindent=0pt
```

のような記述がありました．これにはポイント ‘pt’ という単位が使われています．L^AT_EX において使用できる長さの単位（表 3.4）は色々あります．ポイントは絶対的な長さではないのでクラスファイルによって変わったりプログラムによっても若干の違いがあります．奥村晴彦氏の `jsclasses` ではクラスオプションに `10pt` 以外のフォントサイズ指定がさ

表 3.4 L^AT_EX で使用できる主な単位

単位	読み	補足（数値は概算）	
in	インチ	1 in = 25.4 mm = 72.27 pt	「 」
cm	センチメートル	1 cm = 10 mm = 28.3 pt	「 」
mm	ミリメートル	1 mm = 2.83 pt	「 」
pt	ポイント	1 pt = 0.35 mm	「 」
em	M の字の幅と同じ．	使用中のフォントに依存	「 」
ex	x の字の高さと同じ．	使用中のフォントに依存	「 」
zw	日本語の一文字の幅．	使用中のフォントに依存	「 」

れている場合は紙面の拡大縮小を使っていますので単位がずれます．これには各単位に ‘true’ を付けて長さを指定します．例えば ‘cm’ ならば ‘truecm’ のようにします．

▼ 3.7.2 単位の使い方

単位は基本的に国際単位 SI に従いローマン体，記号はイタリック体で表記します．単位の接頭語として表 3.6 の修飾子が使用できます^{*2}．

表 3.5 SI の基本単位

名称	英語名称	記号	単位	読み	英語読み
長さ	length	<i>l</i>	m	メートル	meter
質量	mass	<i>m</i>	kg	キログラム	kilogram
時間	time	<i>t</i>	s	秒	second
物理量	amount of substance	<i>n</i>	mol	モル	mole
電流	electric current	<i>I</i>	A	アンペア	ampere
熱力学温度	thermodynamic temperature	<i>T</i>	K	ケルビン	kelvin
光度	luminous intensity	<i>I</i>	cd	カンデラ	candela

表 3.6 10^n の修飾子

10^n	10^{12}	10^9	10^6	10^3	10^{-3}	10^{-6}	10^{-9}	10^{-12}
記号	T	G	M	k	m	μ	n	p
名称	テラ	ギガ	メガ	キロ	ミリ	マイクロ*	ナノ	ピコ
英語名称	tera	giga	mega	kilo	milli	micro	nano	pico

* ローマン体のマイクロ (μ) を出力するには textcomp パッケージの `\textmu` コマンドを使います．

数値と単位の間には半角程度の空白を挿入します．`3\,mkg` (3 ミリキログラム) など，修飾子を複数表記してはいけません．`3\,mkg (x)` は正しくは `3\,g` となります．

数値と単位の間には半角程度の空白を挿入します．`3 mkg` (3 ミリキログラム) など，修飾子を複数表記してはいけません．`3 mkg (x)` は正しくは `3 g` となります．

数値と単位の間には半角程度の空白を挿入します．単位とその修飾子はいかなる場合でもローマン体とします．強調部分に単位が含まれる場合でも同様です．

3.8 句読点

句読点 (punctuation) は組み方向を縦書きにするか横書きにするかで違います．レポート・論文の多くは横書きの場合ですから，全角のコンマ‘，’とピリオド‘．’を使うと良いでしょう．ただし，欧文中心の文や段落にはすべて半角の句読点や括弧を使います．

^{*2} この他にも 10^{24} から 10^{-24} まで (Y Z P T G M k m μ n p f a z y) ありますが，実際に用いられるだろう修飾子だけを掲載しました．

The length of a pen should be comrotable
to write with: too long and it makes him
tired; too short and it\ldots.\par
Prof.~Albert Einstein (1897--1955)
was born in German (see fig.~3).

The length of a pen should be comrotable to
write with: too long and it makes him tired;
too short and it...
Prof. Albert Einstein (1897–1955) was born in
German (see fig. 3).

欧文において、コロン、セミコロンなどの記号はコンマ、ピリオドと同様に、記号の前に空白（空き）を入れず、後ろに半角の空白を挿入しています。

丸括弧（パーレン）の左側（起こし）に空白を入れていますが、右側（受け）には入れていません。和文のみの場合は次のようにします。いずれにしても、和文には全角を、欧文には半角の句読点を使うと、問題も少なくなります。

- 句読点は読点‘，’と句点‘。’を使う。
- 単語の引用はかぎ括弧‘`「`’`」`，文の引用はダブルクオート‘`『`’`』`を使う。

3.9 注釈

注釈 (note) とは文章の中で出てきた注意すべき語句を説明するために付けるものです。注釈は読者が読まなくても良い、本文とは関係のない情報を示すために使われます。L^AT_EX では 2 種類の注釈を出力できます。一つはページ下部に出力する脚注 (`\footnote`)，もう一つは注釈語の横に出力する傍注 (`\marginpar`) です。紙面の下端に表示される脚注には `\footnote` 命令を使います。

このように傍注が出力されます

```
注釈語\footnote{注釈内容}
```

レポート・論文の場合、傍注を使わずに脚注のみを使うようにしてください。

この命令を使用すると L^AT_EX は組版時に自動的に `\footnote` で通し番号を付けます^{*3}。脚注の出力は使用しているクラスファイルによって違うので確認してみると良いでしょう。

ラプラス変換やフーリエ変換^a`\footnote{Fourier Translation}`は通常理工系の大学ならば必修で\ldots と思われる。

ラプラス変換やフーリエ変換^aは通常理工系の大学ならば必修で...と思われる。

^a Fourier Translation

3.10 文字の強調

最近のワープロ文書では重要な文字列に下線 (underline) を引いて強調 (emphasis) を表現しているようです。論文や書籍では欧文をイタリック体、和文の場合はゴシック体にします。L^AT_EX では文字列の強調のために `\emph` が使えます。

欧文の強調には `\emph{English Emphasize}` として、和文の強調は `\emph{文字列の強調}` のようにします。

欧文の強調には *English Emphasize* として、和文の強調は文字列の強調のようにします。

^{*3} このように注釈が文章の頁の下端に出力されます。

3.11 特殊記号

アクセント記号などを出力するための特殊文字も用意されており，それらを利用するには表 3.7 の命令を使います．表中のアスタリスク ‘*’ 付きの記号は fontenc パッケージを ‘T1’ というオプション付きで読み込むと出力できます．アクセント類を出力するには表 3.8 の命令を使います．‘i’ と ‘j’ にアクセントを付けるには表 3.7 中の点のない ‘\i’ と ‘\j’ を使います．

表 3.7 特殊記号

å	\aa	ø	\o	†	\dag	Ð	\DJ *	«	\guillemotleft *
Å	\AA	Ø	\O	‡	\ddag	ŋ	\ng *	»	\guillemotright *
æ	\ae	ı	\i	£	\pounds	Ŋ	\NG *	<	\guilsinglleft *
Æ	\AE	J	\j	¡	!‘	þ	\th *	>	\guilsinglright *
œ	\oe	ß	\ss	¿	?‘	Þ	\TH *	„	\quotedblbase *
Œ	\OE	Š	\SS	ð	\dh *			,	\quotesinglbase *
ł	\l	§	\S	Ð	\DH *			"	\textquotedbl *
Ł	\L	¶	\P	đ	\dj *				

表 3.8 アクセント記号

ü	\"u	ā	\={a}	à	\‘{a}	ạ	\d{a}	ǎ	\v{a}	ô	\r{o}
é	\’{e}	ã	\H{a}	á	\b{a}	ơ	\k{o}	ñ	\~{n}		
à	\.{a}	ô	\^{o}	ç	\c{c}	ĩ	\u{ı}	ôo	\t{oo}		

J\org {mu\ss} ein Gel\ande f\"ur
seine Fabrik erwerben.

Jörg muß ein Gelände für seine Fabrik erwerben.

3.12 原稿中での空白の扱い

L^AT_EX では半角スペースとタブはどちらもスペース (white space) として扱われます．二つ以上のスペースが並んでいるときは一つのスペースとして扱われます．また，一つだけの改行もスペースとして扱われます．改行が二つ連続している（空行が存在する）とそれを段落の区切りと判断します．

半角の空白 はこのように 二 つ以上
あっても 一つとみなされます．

空行はこのように段落の区切りになります．
この スペースは一つです．

半角の空白 はこのように 二 つ以上あっても 一つ
とみなされます．

空行はこのように段落の区切りになります．この
スペースは一つです．

3.13 コメントの挿入

ファイルのどの行からでもパーセント ‘%’ があるとそれ以降をコメント (comment out) して扱います。行頭に ‘%’ を置けばそこから行末まですべてがコメントアウトされます。複数行のコメントを挿入したいときは `comment` 環境を使います。これを使用するためには Victor Eijkhout 氏による `comment` パッケージを読み込みます。

```
%\usepackage{comment}
こは出力されますが % こはされない。
\begin{comment}
この環境の中もコメントになるので
\end{comment}
出力されませんか？
```

こは出力されますが出力されませんか？

3.14 ベタ書き

テキストをそのまま出力するときがあると思います。たとえばプログラムリストを載せたいときは特殊記号などが入り、そのままでは記述するのが困難です。そのようなときはベタ書き (verbatim) が可能です。短い文字列の場合は

```
\verb+<文字列>+
```

を使います。複数行になるときは `verbatim` 環境を使います。

```
\begin{verbatim}
ここにベタ書きしたい複数行の文字列を挿入します。
\end{verbatim}
```

<code>\verb #include<stdio.h> は\ldots</code>	<code>#include<stdio.h>は...</code>
<code>\begin{verbatim}</code>	
<code>int main(void){</code>	<code>int main(void){</code>
<code>printf("Hello, World!\n");</code>	<code>printf("Hello, World!\n");</code>
<code>}</code>	<code>}</code>
<code>\end{verbatim}</code>	

`\verb` 命令や `verbatim` 環境にはアスタリスクを付けることができます。さらに `\verb` 命令の場合は (文字列) を括る区切り記号はアスタリスク ‘*’ 以外ならば何でも良いことになっています。

<code>\verb 134 , \verb+456+, \verb9 () 9.\par</code>	<code>134, 456, () . .\par</code>
<code>\verb* 1 3 5 , \verb*9ok? ok?9.\par</code>	<code>1_3_5, ok?_ok? .\par</code>
<code>\begin{verbatim*}</code>	
<code>int main(void){</code>	<code>int_ main(_void_){</code>
<code>printf("Hello, World!\n");</code>	<code>___printf("Hello,_World!\n");</code>
<code>}</code>	<code>}</code>
<code>\end{verbatim*}</code>	

3.15 引用や文の区切り

文献から一文を引用する、段落を引用するという場面があると思います。引用においては「いくつかの単語」、「文」、「段落」、「複数の段落」の四つの引用形態があります。

単語の引用 欧文はシングルクオート ‘ ’ を使い、和文はかぎ括弧 「 」 を使う。

文の引用 欧文はダブルクオート “ ” を使い、和文はかぎ括弧 「 」 を使う。

段落の引用 quote 環境を使い、別段落に組む。複数段落を記述しても、字下げが行なわれない。

複数段落の引用 quotation 環境を使い、別段落に組む。各段落では字下げが行なわれる。

引用の引用 すでに引用している文をさらに引用するならば、欧文は “ ” のようにし、和文は 「 『 』 」 とする。

シングルクオートも2種類あり左シングルクオート (‘) はキーボードの `[Shift]` を押しながら `[@]` を押し、右シングルクオート (’) は `[Shift]` を押しながら `[7]` を押すと入力できると思います。LaTeX ではこれらを区別して記述します。絶対に `[Shift]+[2]` を押してダブルクオート “ ” で引用符を代用してはいけません。

文の引用ではダブルクオートを使います。Wordなどでダブルクオートを挿入すれば自動的に“一文”のように変換されますが LaTeX ではシングルクオートをうまく組み合わせで記述します。これは左シングルクオートを二つと右シングルクオートを二つで括ることになります。他に1文用の quote 環境や段落ごと引用するための quotation 環境があります。

‘単語はシングルクオートで囲む’
 ‘‘文はダブルシングルクオートで囲む’’

さらに段落ごと引用する場合は段落の左側を字下げして出力します。場合によっては文字を小さくします。一つの段落だけを引用する場合は quote 環境を、複数の段落を引用するならば quotation 環境を使います。

`\begin{quote}` 段落引用は quote 環境で囲む`\end{quote}`
`\begin{quotation}` 段落引用は quotation 環境で囲む`\end{quotation}`

一般的に以下のような使い方になります。

‘単語’の引用はシングルクオートで‘‘文章の一文’’の引用は左シングルクオート二つと右シングルクオート二つです。“ダブルクオート”で引用符を表してはいけません。

‘単語’の引用はシングルクオートで“文章の一文”の引用は左シングルクオート二つと右シングルクオート二つです。“ダブルクオート”で引用符を表してはいけません。

段落を引用する quote 環境の他にも

```
\begin{quote} 行頭の字下げをする
  段落引用の quotation 環境がある。
\end{quote} といわれている。
```

段落を引用する quote 環境の他にも

```
行頭の字下げをする段落引用の quotation
環境がある。
といわれている。
```

和文の引用における引用符は全角のかぎ括弧「」を使い、欧文の場合の引用符には半角のクオート“”を使います。和文の引用の中の引用には二重括弧を用います。和文の場合、括弧の中に句点を入れてはいけません。

“‘\,FUN: Future University-Hakodate’”
は恐らく‘FUNNIST’との密接な関わりがあり、
渡辺によると「未来らによると『FUNNISTはFUNに
ある組織である』という説がある」と考察して
いる。

“FUN: Future University-Hakodate”は恐らく
‘FUNNIST’との密接な関わりがあり、渡辺によ
ると「未来らによると『FUNNISTはFUNにある組
織である』という説がある」と考察している。

▼ 3.15.1 書籍名や雑誌名の引用

書籍名や雑誌名を引用する場合はその名前をイタリック体にします。欧文の場合は`\emph` 命令を使います。和文の書籍名を引用する場合は二重かぎ括弧『』を、雑誌名を引用する場合はかぎ括弧「」を使います。

`\emph{article's Name}` (欧文の場合)

『書籍名』(和文の書籍)

「雑誌名」(和文の雑誌)

以上のような方法を使って何か別の文書を示す場合はその文書名を強調表示します。

渡辺が2004年に`\emph{Natural}`に投稿した論文
「論文作成のいろは」は未来出版から『論文作
成の手引き』に改題されて出版されている。

渡辺が2004年に *Natural* に投稿した論文「論文作
成のいろは」は未来出版から『論文作成の手引き』
に改題されて出版されている。

▼ 3.15.2 ダッシュ

ダッシュには和文と欧文のものを併せると4種類ほどあります。ひとまとめにしたい単語の区切りや、文の中断などに使います。

en-dash ‘-’ 数値の範囲などを表します。和文の場合は波ダッシュ‘~’を使う例も見られ
ますが、「10~30人」という表記は避けた方が無難です。

em-dash ‘—’ 文の中断を表します。

全角ダッシュ‘ ’ 欧文の en-dash に近い意味を表しますが、若干高さが違います。

倍角ダッシュ‘ ’ 和文での文の中断などを表します。

さらにダッシュに似たものにハイフンとマイナスがあります。

ハイフン‘-’ 欧文で単語の途中にハイフネーションとして挿入される。

マイナス‘-’ 数学記号で負の数値を表す。

以上の記号を混同することなく正しく使うのが好ましいです。倍角ダッシュを出力するためには `okumacro` パッケージを読み込みます。出力方法は表 3.9 の通りです。

“‘When I was a dog---a big dog---I could
read about 100--200 books in a day.
This is a just fairy-tale.’”\par

“When I was a dog—a big dog—I could read
about 100–200 books in a day. This is a just
fairy-tale.”

表 3.9 ダッシュなど

記号の種類	出力	入力・命令	
en-dash	–	--	ハイフンを二つ
em-dash	—	---	ハイフンを三つ
全角ダース			全角のダッシュ
倍角ダース		\	‘\’ と全角ダース二つ
ハイフン	-	-	そのまま
マイナス	—	\$-\$	数式中でハイフン一つ

通常ハイフンやダッシュの両隣には空白を入れません。ハイフンによって単語を一塊にしている語句は、ハイフンの途中で改行してはいけません。これは通常の 1 単語のハイフネーションと重複する可能性があるからです。

`{\TeX}\mbox{for-each}`文は Perl における `\mbox{foreach}`文とは性質が異なるため、`\mbox{X-ray}`の影響を受けた Future `\mbox{University-Hakodate}`は `\mbox{if-then}`文を使う傾向にある。

\TeX の `for-each` 文は Perl における `foreach` 文とは性質が異なるため、`X-ray` の影響を受けた Future `University-Hakodate` は `if-then` 文を使う傾向にある。

▼ 3.15.3 改行

改行 (line break) はバックスラッシュ ‘\’ (Windows などでは円 ‘¥’) を二つ並べて ‘\\’ のようにすれば入れることが可能ですが、文章の中に改行を入れるときは慎重に挿入しなければいけません。できることならばユーザ側の強制的な改行は挿入しないほうが良いでしょう。同じ段落とある文字列を区別したいときは改行ではなく引用 (3.15 節参照) を使うとうまく行くことが多いです。

```
\\*[⟨長さ⟩]
\\newline
\\par
```

任意引数に改行を行うときの縦の長さを指定できます。ページの先頭での改行を行うことはできません。アスタリスクを付けると改行直後にページを改めることを禁止します。`\\newline` は ‘\\’ とほぼ同時の命令です。`\\par` は改行ではなく改段落、すなわち段の終わりを示します。その直後の文字列は字下げされます。

改行は `\verb|\\|` のように `\\` バックスラッシュを二つ続けて書くと `\\[1cm]` ユーザによる強制的な改行が挿入されます。`\\par` この文章は新しい段落から組まれ `\\newline` 字下げされる場合があります。

改行は `\\` のように
バックスラッシュを二つ続けて書くと

ユーザによる強制的な改行が挿入されます。
この文章は新しい段落から組まれ
字下げされる場合があります。

3.16 空白について

空白は要素と要素を区切るために使われます。空き (space) の広さによって意味が違います。正しい量の空白を挿入しなければ意味が変わってしまいます。

3

▼ 3.16.1 文章の中の空きの調整

まず一つの段落内における空白の種類を考えてみましょう。日本語の場合はある文字とそれに隣接する文字のあいだに挿入される文字間空白 (字間) というものが存在します。漢字と漢字がぎゅうぎゅうに詰められていては、非常に読みづらいでしょう。この処理は通常日本語 $\text{T}_{\text{E}}\text{X}$ が自動的に行います。欧文でもこれは知らないあいだに処理されています。例えば合字 (ligature) や字詰め (kerning) などと呼ばれるものがあります。以下の入出力を見比べてください。

```
The files were found in a folder and
were shuffled by anyone.\par
The f{}iles were found in a folder and
were shuf{}f{}led by anyone.
```

```
The files were found in a folder and were shuffled
by anyone.
The files were found in a folder and were shuffled
by anyone.
```

ここでは ‘fl’ や ‘ffl’ などがある例です。

欧文の場合、単語と単語のあいだに空白を挿入します。これを単語間空白と呼びます。これは人間が意図的に単語の区切りとして ‘My_name_is_Thor.’ のように挿入します。

さらに文と文とを区切るための文間空白があります。これは文の終わりを示すもので、単語間空白や文字間空白よりも広い空白になります。 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ では

- ピリオドの前の文字が大文字ならば単語間空白を挿入する。
- ピリオドの前の文字が小文字ならば文間空白を挿入する。

という二つのルールしか持っていません。

そこで問題になるのが大文字で終わる単語や小文字を含む文です。

```
I want to be a Mr. Right and go to
N.Y. I wish I could. D.~E. Knuth
means ‘Donald Ervin Knuth.’
```

```
I want to be a Mr. Right and go to N.Y. I
wish I could. D. E. Knuth means ‘Donald Ervin
Knuth.’
```

‘Mr.’ と ‘Right’ のあいだの空白のほうが ‘N.Y.’ と ‘I’ のあいだの空白よりも若干広くなります。地名の ‘N.Y.’ の場合、ピリオドに空きはありますが、人名の ‘D. E.’ には空きを入れます。これらを正しい空白にするためには人間が明示的に二つの命令を使います。単語間空白を挿入するためには `_` 命令を、文間空白には `\@` 命令を使います。

```
I want to be a Mr. Right and go to
N.Y. Let me do.\par
I want to be a Mr.\ Right and go to
N.Y\@. Let me do.
```

```
I want to be a Mr. Right and go to N.Y. Let
me do.
I want to be a Mr. Right and go to N.Y. Let
me do.
```

そして行と行のあいだの行間空白（行間）がありますし、段落と段落のあいだの段落間空白もあります。これらは \LaTeX が最適な空白の量を調節してくれているので、普段は気にすることはないでしょう。

最後に文章における空白をまとめると次の五つがあるということです。

文字間空白 (**letter space**) 文字間に挿入される空白。

単語間空白 (**word space**) 単語間に挿入される空白。 $\backslash_$ 命令で明示的に挿入できる。

文間空白 (**sentence space**) 文間に挿入される空白。 $\backslash_$ 命令で明示的に挿入できる。

行間空白 (**leading**) 行間に挿入される空白。

段落間空白 (**paragraph skip**) 段落間に挿入される空白。 \backslashpar 命令で明示的に段落の終了を告げることができる。

▼ 3.16.2 その他注意すること

それらが並んでいることで一つの意味を持つ単語間には改行を入れないようにします。例えば人名やページ番号、略語などは、一まとめにします。これにはチルダ ‘ \sim ’ を使います。

Mr. \sim Sato read page \sim 10 and looked at
figure \sim 3 and table \sim 2 in the book.

Mr. Sato read page 10 and looked at figure 3
and table 2 in the book.

引用符が隣り合うときには、引用符と引用符のあいだに小さな空白を挿入します。

“‘Hello’ is a greeting and I always
say ‘Hello.’” \backslashpar “‘Hello’ is a
greeting and I always say ‘Hello.’”

“‘Hello’ is a greeting and I always say ‘Hello.’”
“‘Hello’ is a greeting and I always say ‘Hello.’”

▼ 3.16.3 和文と欧文のあいだの空白

日本語 \TeX では和文と欧文のあいだには空白が挿入されています。これを和文と欧文の四分空きと呼びます。四分空きとは全角空白の $\frac{4}{16}$ の空白のことです。これは和文組版の規則で挿入すべき空白であって、挿入したほうが美しく見えると言われています。以下の例を見ると良く分かるでしょう。例では \backslashmbox で四分空きを無効にしています。

日本語と English のあいだには四分空きが
ないと $\backslashmbox\{\}$ English $\backslashmbox\{\}$ ではない。

日本語と English のあいだには四分空きが
ないと English ではない。

普段は何も意識せずに空白が挿入されているので問題ないのですが、原稿の記述の仕方によってその空白が四分空きよりも広くなります。意図的に全角文字と半角文字のあいだに半角空白を挿入するとその部分は四分空きよりも広い単語間空白になるとときがあります。組版の規則に従うとこの空白は統一すべきですので入力段階でそれらに気を付けます。例として \LaTeX という記号と全角文字の書き方を示します。日本語 \TeX は自動的に隣り合う文字が半角文字か全角文字かを判別してくれます。始めは日本語 \TeX にその処

理を任せて、慣れてきたら自分でその空白を調節すればよいでしょう。実際に入力して試してください。

<code>\LaTeX と日本語\TeX</code>	<code>\\</code>	\LaTeX と日本語 \TeX
<code>\LaTeX\ と日本語 \TeX</code>	<code>\\</code>	\LaTeX と日本語 \TeX
<code>{\LaTeX}と日本語{\TeX}</code>	<code>\\</code>	\LaTeX と日本語 \TeX
<code>{\LaTeX} と日本語 {\TeX}\\</code>		\LaTeX と日本語 \TeX

使っている欧文書体の種類によっても違いますし、好みもあるのでこれだと断言できませんが、入力するうえでの作業を考えると三つ目が一番手軽だと思います。ただし、この方法をとるときは欧文同士の空白に注意します。

<code>\TeX and {\LaTeX}are very famous.\\</code>	\TeX and \LaTeX are very famous.
<code>{\TeX} and {\LaTeX} are very pupular.</code>	\TeX and \LaTeX are very pupular.

入力ファイルでは‘`\TeX`’の後に空白を挿入しているつもりでも、出力において空白は`\TeX`に吸収されてしまいます。

3.17 箇条書き

箇条書き (itemization) には三つの環境を使うことが出来ます。

`itemize` 環境 項目の先頭に記号 (ラベル) が付く記号付き箇条書き環境。環境の深さによって記号が‘●, −, *, ’のように自動的に変わる。

`enumerate` 環境 項目の先頭に通し番号が付く番号付き箇条書き環境。深さによって通し番号が‘1, (a), i, A’のように自動的に変わる。

`description` 環境 項目の前に説明を `\item` の任意引数で指定する説明付き箇条書き環境。

レポートや論文の場合はなるべく箇条書きは避けて、文章による記述が望ましいようです。理解のしやすさを考えれば箇条書きを使うべきでしょう。これらの環境は入れ子 (nest) にすることが可能です。入れ子に出来る項目の深さは通常四つまでです。`itemize` 環境の先頭の記号は入れ子にした場合自動的に変更されます。各環境においての項目は `\item` 命令を使います。`itemize` においては `\item[\#]` とすることで先頭のラベルの記号を指定することが可能です。

<code>\begin{itemize}</code>	
<code>\item 入れ子にしたい。</code>	● 入れ子にしたい。
<code>\item[*] 入れ子になる。</code>	* 入れ子になる。
<code>\begin{itemize}</code>	
<code>\item 入れ子です。</code>	− 入れ子です。
<code>\end{itemize}</code>	
<code>\end{itemize}</code>	

```

\begin{enumerate}
  \item はじめの項目.
  \item 次の項目.
\begin{description}
  \item[項目 1] 説明 1.
  \item[項目 2] 説明 2.
\end{description}
\end{enumerate}

```

1. はじめの項目.
 2. 次の項目.
- 項目 1 説明 1.
- 項目 2 説明 2.

▶ 問題 3.1 ここまでの節を読んだだけでも、ある程度の規模の文書を作成することが出来ます。次のような `<file>.tex` を作成し、実際にタイプセットを行い、その出力結果を吟味してください。ただし、目次も出力するためには2回ほどタイプセットを行ないます。

```

\documentclass[a4j]{jsarticle}
\title{はじめての\LaTeX}% 題名
\author{自分の名前}% 著者
\date{\today}% 日付
\begin{document}% 本文
\maketitle% 表紙
\tableofcontents% 目次
\section{節見出し}% 節見出し
節見出しは\verb|\section|コマンドを使います。
\subsection{小節見出し}% 小節見出し
小節見出しは\verb|\subsection|を使います。
%
\section{文章の記述}
この節では文章の記述について論じます。
\subsection{引用}
一文を引用する場合はカギ括弧を使います。一説によると
「カギ括弧は引用に使う」と言われている。
段落ごと引用するということは次のようになっている。
\begin{quote}
段落ごとの引用の場合は\verb|quotation|環境を使い、\emph{行頭}
を字下げしない。複数段落の引用の場合は\verb|quotation|環境
を使い、行頭を字下げする。
\end{quote}
\subsection{箇条書き}
箇条書きには以下の三つが用意されている。
\begin{description}
\item[記号付箇条書き] ラベルの先頭に記号がついた箇条書き。
\item[番号付箇条書き] ラベルの先頭に番号がついた箇条書き。
\item[説明付箇条書き] ラベルの先頭に説明がついた箇条書き。
\end{description}
\end{document}

```

この入力の実出力例は図 3.1 のようになります。

はじめての L^AT_EX

自分の名前

2004 年 9 月 20 日

目次

1	節見出し	1
1.1	小節見出し	1
2	文章の記述	1
2.1	引用	1
2.2	箇条書き	1

1 節見出し

節見出しは `\section` コマンドを使います。

1.1 小節見出し

小節見出しは `\subsection` を使います。

2 文章の記述

この節では文章の記述について論じます。

2.1 引用

一文を引用する場合はカギ括弧を使います。一説によると「カギ括弧は引用に使う」と言われている。段落ごと引用するということは次のようになっている。

段落ごとの引用の場合は `quotation` 環境を使い、行頭を字下げしない。複数段落の引用の場合は `quotation` 環境を使い、行頭を字下げする。

2.2 箇条書き

箇条書きには以下の三つが用意されている。

- 記号付箇条書き ラベルの先頭に記号がついた箇条書き。
- 番号付箇条書き ラベルの先頭に番号がついた箇条書き。
- 説明付箇条書き ラベルの先頭に説明がついた箇条書き。

図 3.1 テキスト入力の出力例

3.18 行揃え

行揃え (flushing) には三つの環境と三つの宣言型コマンドを使うことができます (表 3.10)。環境型のコマンドは広い範囲に使い、宣言型のコマンドは一つの要素や別の環

表 3.10 揃えの命令と宣言

種類	環境	宣言
左揃え	<code>flushleft</code>	<code>\raggedright</code>
中央揃え	<code>center</code>	<code>\centering</code>
右揃え	<code>flushright</code>	<code>\raggedleft</code>

境の中で使うことができます。中央揃え (centering) には `center` 環境です。1 行もしくはそれ以上の文字列、表、図などを中央に寄せることが可能です。行頭や最終行に改行は入れません。右揃え (flushright) には `flushright` 環境です。文字列を右寄せにします。左揃え (flushleft) には `flushleft` 環境です。字下げを行わずに左に寄せます。

ビジネス文書で大活躍するでしょう。

```
\begin{flushleft}
```

段落の字下げを行わずに \\

文字列を左に揃えます。

```
\end{flushleft}
```

ビジネス文書で大活躍するでしょう。

```
\begin{center}
```

文章を\\ 中央揃えに \\ します。

```
\end{center}
```

ビジネス文書で大活躍するでしょう。

```
\begin{flushright}
```

ビジネス文書で活躍中の\\

`flushright` 環境です。

```
\end{flushright}
```

ビジネス文書で大活躍するでしょう。

段落の字下げを行わずに

文字列を左に揃えます。

ビジネス文書で大活躍するでしょう。

文章を

中央揃えに

します。

ビジネス文書で大活躍するでしょう。

ビジネス文書で活躍中の

`flushright` 環境です。

この三つの行揃えのコマンドを使ってビジネス文書に良く見られる書式を作成できます。

```
\begin{flushright}
```

緊急連絡 \\ 2005 年 3 月 31 日

```
\end{flushright}
```

```
\begin{flushleft}
```

渡辺 徹殿

```
\end{flushleft}
```

```
\begin{flushright}
```

未来会社\\ 人事課

```
\end{flushright}
```

```
\begin{center}
```

```

人事異動のお知らせ
\end{center}
あなたは 2004 年度から檜山方面に配属されます .
\begin{flushright} 以上 \end{flushright}

```

	緊急連絡
	2004 年 3 月 31 日
渡辺 徹殿	
	未来会社 人事課
人事異動のお知らせ	
あなたは 2004 年度から檜山方面に配属されます .	
	以上

3.19 書体について

文字 (character) は意思伝達手段であって、長いあいだに洗練された媒体です。怒りの意思を強く込めたいならば人は荒々しく文字を書くでしょうし、優しさを込めたいならば丸みを帯びた書き方になるでしょう。以上のような文字の形を書体 (typeface) と呼びます。

世の中にはこれらを書体というひとつの枠組みで整理しています。書体は読者に対して何らかのメッセージを分かりやすく伝えるために変更される場合があります。ですから書体を変更するということには必ず意味があるべきなのです。むやみやたらに書体を変更しても逆に読者を混乱させます。また自分だけのルールで書体を変更しても読者には何の意味なのかが分かりませんので、一般的に使われている書体に関するルールを守るのもマナーです。

L^AT_EX はマークアップ型のシステムなのでユーザーが直接書体変更用の命令を使うことは本来ならば必要のないことだと思われます。以下のコマンドは直接使うのではなく新規に環境を定義して用いるのが望ましいでしょう。

▼ 3.19.1 文字の大きさの変更

L^AT_EX においては比較的簡単に文字の大きさを変えることが可能ですが、文字は文書クラスオプションで指定した基準の文字の大きさに応じて変更されます。文字の大きさを変更したいときは表 3.11 の宣言型のコマンドを

```
{\命令 文字の大きさを変えたい文字列}
```

のように使用します。

表 3.11 文字の大きさの変更

大きさ	命令	出力例
とても小さい	<code>\tiny</code>	野鳥
かなり小さい	<code>\scriptsize</code>	花鳥
小さい	<code>\footnotesize</code>	雷鳥
やや小さい	<code>\small</code>	白鳥
普通	<code>\normalsize</code>	飛鳥
やや大きい	<code>\large</code>	やちょう
大きい	<code>\Large</code>	かちょう
かなり大きい	<code>\LARGE</code>	らいちょう
とても大きい	<code>\huge</code>	はくちょう
特大	<code>\Huge</code>	ひちょう

表 3.12 基準の文字の大きさによるコマンドの挙動の違い

コマンド\基準の大きさ	10 pt	11 pt	12 pt	使用すべき要素 *
<code>\tiny</code>	5 pt	6 pt	6 pt	振り仮名
<code>\scriptsize</code>	7 pt	8 pt	8 pt	
<code>\footnotesize</code>	8 pt	9 pt	10 pt	索引・脚注
<code>\small</code>	9 pt	10 pt	11 pt	図表見出し
<code>\normalsize</code>	10 pt	11 pt	12 pt	小小節見出し・本文
<code>\large</code>	12 pt	12 pt	14 pt	小節見出し
<code>\Large</code>	14 pt	14 pt	17 pt	節見出し
<code>\LARGE</code>	17 pt	17 pt	20 pt	
<code>\huge</code>	20 pt	20 pt	25 pt	部・章見出し番号
<code>\Huge</code>	25 pt	25 pt	25 pt	部・章見出し

* 使用すべき要素は1段組での場合です。

そういえば、`{\scriptsize これ}`は小さい文字だけど、`{\Large こっち}`は大きい文字になってるね。

そういえば、これは小さい文字だけど、**こっち**は大きい文字になってるね。

このような書体の大きさを変更するコマンドを直接使うのは好ましくなく、きちんとマークアップ付けをするべきです。例えば強調のために文字を大きくしたいのであれば新規に `\kyocho` 命令を作ります。

```
\newcommand{\kyocho}[1]{\Large#1}
\newcommand{\Kyocho}[1]{\LARGE#1}
ああそういえば\kyocho{ここは大事だからね} .
それに\Kyocho{ここはもっと大事}だよ .
```

ああそういえばここは大事だからね .
それにここはもっと大事だよ .

▼ 3.19.2 書体の変更

LaTeX において書体の種類は次の四つに分けられます．サイズに関しては前述の通りです．

ファミリー デザイン上の系統の種類．
シリーズ 線の太さと文字幅の違いによる種類．
シェイプ 形状の変化の違いによる種類．
サイズ フォントの大きさ．

ローマンファミリーは本文の書体に使います．サンセリフファミリーは見出しなどに使うべきですが，近年の公式な文書においてもサンセリフファミリーが本文に使われることもあります．タイプライタファミリーはプログラムのコードを示す場合に使われます．

表 3.13 の一覧から適切な書体を選んでください．ファミリーとシリーズとシェイプは

表 3.13 書体を変更するコマンド

種類	命令	宣言	出力
ローマンファミリー	<code>\textrm</code>	<code>\rmfamily</code>	ABCabc
サンセリフファミリー	<code>\textsf</code>	<code>\sffamily</code>	ABCabc
タイプライタファミリー	<code>\texttt</code>	<code>\ttfamily</code>	ABCabc
ミディアムシリーズ	<code>\textmd</code>	<code>\mdseries</code>	ABCabc
ボールドシリーズ	<code>\textbf</code>	<code>\bfseries</code>	ABCabc
イタリックシェイプ	<code>\textit</code>	<code>\itshape</code>	<i>ABCabc</i>
スラントシェイプ	<code>\textsl</code>	<code>\slshape</code>	<i>ABCabc</i>
スモールキャピタルシェイプ	<code>\textsc</code>	<code>\scshape</code>	ABCABC

それぞれ組み合わせて使うことが出来ます．例えば「セリフがなくて太いフォント」という文字を出力したければ次のようにします．

```
\textsf{\textbf{Typeface}}
{\sffamily\bfseries Typeface}
```

Typeface Typeface

使用している基本書体によっては出力できないタイプもあります．

```
\texttt{\textit{Typewriter bold
extended?}} \textsc{Small Caps}.
\textit{\textbf{Bold italic}}.
{\ttfamily \itshape typewriter
bold extended.}
```

Typewriter bold extended? SMALL CAPS.
Bold italic. typewriter bold extended.

書体のファミリーやシェイプなどを先に指定してから大きさを変更します．

`{\Large\textbf Large Bold?}` 成功．
`{\textbf\Large Bold Large?}` 失敗．

Large Bold? 成功．
Bold Large? 失敗．

和文の書体は基本的には明朝体とゴシック体の二つしか用意されていません (表 3.14)．これは従来の和文組版で二つの書体しか使われなかった名残です．現在の $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ で和文の多書体を図ることはそれ程難しくありません．ただ不用意に和文を多書体にしても読者がそれに慣れていないと思われますので，悪戯に行わないほうが良いかもしれません．

表 3.14 和文書体のファミリー

種類	命令	宣言	出力
明朝ファミリー	<code>\textmc</code>	<code>\mcfamily</code>	永字八法とは何ですか？
ゴシックファミリー	<code>\textgt</code>	<code>\gtfamily</code>	永字八法とは何ですか？

和文組版において明朝体は通常の文章の組版，ゴシック体は`\textgt{文章の強調に}`使われます．`{\gtfamily` 見出しも強調すべき要素なのでゴシック体にするのが普通です．}

和文組版において明朝体は通常の文章の組版，ゴシック体は文章の強調に使われます．見出しも強調すべき要素なのでゴシック体にするのが普通です．

3.20 文章の修正

このようにして基本的な文章の論理構造を組み上げて，結果的に紙の上などに出力するわけですが，一発で完璧な文書になることはほとんどありません．何度も修正と加筆を繰り返し，最終的な論文に仕上がるものと思います．

そのときに必要なのは文章の校正に関わる約束事です． $\text{L}\text{A}\text{T}\text{E}\text{X}$ ではほとんどの多くの処理を半自動的に行うので，普段は気にならない部分です．例えば半角の英数字と全角の日本語とのあいだには四分空きといって，全角空白の 4 分の 1 のスペースを挿入したり，行の先頭に句読点があってはいけないという，行頭禁則処理の問題も $\text{L}\text{A}\text{T}\text{E}\text{X}$ ($\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ において) は半自動で行います．

このような自動的な処理以外にもユーザー側の入力ミスにより修正が必要になる場合があります．その場合は 1 度作成した文章を校正記号 [6] などを使って修正するのが良いでしょう．

現在では文章はコンピュータ上ですべて組むことが出来るので，間違いを見つけたらその場ですぐに修正可能です．紙に印刷してチェックするという作業は非効率のかもしれませんが．コンピュータのモニター上と印刷した紙上の両者の特性を活かして文章を修正してください．文章作成 [20, 19, 11] 上で注意すべき点として

- 1 文を長くしすぎていないか．
- である調で統一されているか．
- 修飾語の関係をはっきりしているか．

- 同音異義語などの間違いはないか．
- 段落の区切り，章の区切りは明確か．

などが挙げられます．

3.21 クラスとパッケージ

3

L^AT_EX はマークアップ言語なので書式と内容は分離されるのが普通です．そこでクラス (class) とパッケージ (package) という二つのファイルを使うようになっています．

L^AT_EX では文書の書式を決定するためにクラスというものを宣言します．クラスはドキュメントクラスとか文書クラスなどと呼ばれています．また，便利な機能を集めたものをパッケージと呼びます．パッケージはマクロパッケージとか，ただ単にマクロなどと呼ばれます．

そうして，L^AT_EX の原稿 (ソースファイル) では必ず文書の先頭に

```
\documentclass[〈オプション ,...〉]{〈クラス〉}
\usepackage[〈オプション ,...〉]{〈パッケージ〉}
```

のような記述をして，文書の書式を大雑把に決定します．

例えば，本文が日本語で画像を含み，書体の大きさが 11 ポイントで二段組の記事を書こうと思えば

```
\documentclass[twocolumn,11pt]{jarticle}
\usepackage{color}
\usepackage{graphicx}
```

のように原稿中で宣言します．使用するクラスの中にはオプションが存在し，上記のように 2 段組のために *twocolumn* やフォントの大きさを指定するために *11pt* というオプションを指定します．また衝突の起きない限り，複数のパッケージを使うことを同時に宣言することもできます．

```
\documentclass[twocolumn,11pt]{jarticle}
\usepackage[dvips]{graphicx,color}
```

クラスとパッケージを明確に区別するためにクラスの拡張子には *.cls* を，パッケージの拡張子には *.sty* を付けるようにしています．

▼ 3.21.1 標準的なクラス

L^AT_EX や pL^AT_EX の範囲内で提供されている標準的なクラスを紹介します．クラスファイルは *〈classes〉.dtx* と *〈classes〉.ins* という二つのファイルで配布されることが多いようです．

日本語を含まないような文書には欧文専用のクラスが使用できます．それぞれどのような文書を作成したいかによって何を用いるかが分かります．標準では *article*，*report*，*slides*，*proc* が使えます．

日本語の文書では、標準で以下のクラスが使えます。

jarticle 小規模の日本語の記事を作成するためのクラス。

jreport 日本語の報告書を作成するためのクラス。

jbook 日本語の書籍を作成するためのクラス。

以上の jarticle, jreport, jbook の三つをまとめて jclasses と呼ぶことがあります。

▼ 3.21.2 クラスオプション

ドキュメントクラス（文書クラス、または単にクラス）にはもう少し詳細な設定を行うことができます。 \documentclass の任意引数として記述します。多くのクラスファイルでは次のクラスオプションが使えると思います。

文字サイズ $\langle 10pt, 11pt, 12pt \rangle$ 原稿で基本となる文字の大きさを決めます。この文字サイズを基準としてさまざまなパラメータが設定されます。標準は 10pt。

用紙サイズ $\langle a4paper, a5paper, b5paper, letterpaper \rangle$ 原稿の用紙の大きさを指定します。和文の場合はこの他に *b4paper, a4j, a5j, b4j, b5j* などです。geometry パッケージや jsclasses を使うと選択の幅が広がります。

用紙方向 $\langle landscape \rangle$ 用紙を横置きにします。標準は縦置きです。

印刷面 $\langle oneside, twoside \rangle$ 用紙の片面 (*oneside*) だけに印刷するかそれとも両面 (*twoside*) に印刷するかを指定します。

段組 $\langle onecolumn, twocolumn \rangle$ 一段組 (*onecolumn*) にするか二段組 (*twocolumn*) にするかを指定します。

表題 $\langle titlepage, notitlepage \rangle$ 表題を独立して出力する (*titlepage*) か、同じページに出力する (*notitlepage*) かという表題のレイアウトを指定します。

数式の位置 $\langle fleqn \rangle$ 別行数式の位置を左揃えに指定します。標準は中央揃えです。

数式番号の位置 $\langle leqno \rangle$ 数式番号の位置を左側に指定します。標準は右側です。

ドラフト $\langle draft, final \rangle$ 文書の領域をはみ出してしまった箇所に印をつけるかどうか。執筆途中で印刷するときにはドラフトモードにする。ドラフトモードの *draft*、原稿が完成したら *final* に変更する。標準は *final*。

左右起し $\langle openright, openany \rangle$ (j)report や (j)book において章などの開始ページの指定をする。常に奇数ページで起こす (*openright*) か、どちらからでも起こす (*openany*) かを設定する。(j)report の標準は *openany*。(j)book の標準は *openright*。

最近では、奥村晴彦氏が管理している jsclasses というクラスファイル群が定評です。このクラス群を導入すると、

jsarticle 小規模の日本語の記事を作成するためのクラス。

jsbook 日本語の書籍や報告書を作成するためのクラス。

jspf 某学会誌用のクラス。

の三つが使用できます。これらのクラスで指定できるクラスオプションが `jclasses` に追加されています^{*4}。以上の `jsarticle`, `jsbook`, `jspf` の三つをまとめて `jsclasses` と呼びます。

▼ 3.21.3 標準で利用できるパッケージ

\LaTeX を導入すると一緒に添付される標準的なパッケージがあります。これらはプリアンブル部分に

```
\usepackage[<オプション>]{<パッケージ>}
```

として使用可能になります。各パッケージの詳細な説明書が読みたいときは

```
$ platex filename.dtx
```

とすれば `<filename>.dvi` が作成されます。索引の作成や目次の作成、相互参照の解決などをすれば完全な DVI ファイルが完成します。各ソースファイルへの検索パスがなければ該当する `<filename>.dtx` を検索することはできません。Windows ならばファイルの検索、Unix 系 OS ならば `find` コマンドなどで探してください。大抵は \LaTeX をインストールしたディレクトリ（フォルダ）の下 '`$texmf/tex/latex/base`' にあります。

\LaTeX がコンピュータに導入されているならば以下の応用的なマクロやソフトウェアが同封されていることでしょう。これらのファイルは欧文の文書を作成するうえでは必須のものとなっています。日本語の文書のみを作成するならば、いくつかのマクロやソフトウェアは必要ないでしょう。

\LaTeX \LaTeX 米国数学会（American Mathematical Society）が提供しているソフトウェア並びにパッケージ。 \LaTeX という \TeX 用を \LaTeX でも使えるようにしたもの。マクロ、フォントなどを総称した呼び名が \LaTeX で、パッケージの名前は `amsmath` と言う。数学系の文書を書くときには必須のマクロ。

babel 多言語を \LaTeX で扱うためのマクロ。このマクロを日本語と共存させるためには少々工夫が必要。

graphicx 画像の挿入や加工などを担うマクロ。同時に `color` というマクロも含まれる。これはデバイス（装置）依存の機能で環境により出力が異なることがある。

tools \LaTeX 3 プロジェクトチームによって提供される標準からは外されたマクロ。

これらのマクロについては少なくとも『 \LaTeX コンパニオン』か『文書処理システム \LaTeX 2_ε』に記述されていることが保証されています。

\LaTeX 3 プロジェクトチームによって提供される `tools` は '`$texmf/tex/latex/tools`' に置かれており、その内訳は以下のとおりです。

array `array` や `tabular`, `tabular*` のような表や行列を拡張した環境を使うことができるマクロ。

calc \LaTeX での計算を楽にするマクロ。

^{*4} (j)classes で定義されていたいくつかのクラスオプションが実装されていません。

dcolumn	表や行列の環境で小数点などを揃えるためのマクロ。
delarray	行列で括弧付けを容易にするためのマクロ。
hhline	表や行列で複雑な罫線を簡単に引くことができるマクロ。
longtable	ページをまたぐような、非常に長い表を作るときに使うマクロ。
tabularx	通常の tabular 環境よりも幅に関して柔軟な表を作るためのマクロ。
afterpage	\clearpage の拡張版のような\afterpage が使えるマクロ。
bm	数式中で太字を簡単に使うようにするためのマクロ。
enumerate	enumerate 環境を拡張するためのマクロ。
ftnright	2 段組で全ての脚注を右側に表示するマクロ。
indentfirst	jarticle や jreport などの標準的なクラスで、章(\chapter)や節(\section)の直後の段落でも字下げを行うようにするマクロ。通常は字下げしないように設定されているので、和文文書を作成しているときはいつでも読み込むようにすれば良い。
layout	現在の文書のページレイアウトを表示するマクロ。
multicol	多段組を実装するためのマクロ。
showkeys	\label, \ref, \cite などの相互参照のラベル名 (keys) を表示するためのマクロ。
theorem	定理型環境を簡単に宣言するためのマクロ。
varioref	indind 相互参照の簡略化相互参照をしやすくするためのマクロ。
verbatim	verbatim 環境を拡張するためのマクロ。
xr	別の文書とでも相互参照できるようにするためのマクロ。
xspace	文中で使われるようなマクロに適切な空白の挿入などを行うマクロ。

第 4 章

参考文献の出力

論文などの文書で重要なのが参考文献です．参考文献の扱いがきちんとできればより良い論文になります．参考文献を明記することはその文献の著者に対する礼儀です．さらに読者がその論文に興味を持ったとき，その事項を深く知るための道しるべにもなります．そもそも他人の著作物を（転載ではなく）引用するには著作権法という法律の範囲内で行う必要があります．この章では $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ での参考文献の取り扱い方を紹介します．

4.1 参考文献の明記

参考文献 (references) を明記することはその文献の著者に対する礼儀です．さらに読者がその論文に興味を持ったとき，その事項を深く知るための道しるべにもなります．そもそも他人の著作物を（転載ではなく）引用するには著作権法という法律の範囲内で行う必要があります．

参考文献は，文書の巻末にまとめて記載するものや，脚注としてそのページに記載する書式などがあります．本文中では括弧書きで「著者名と年号」だけの表示にしたり，参考文献の通し番号だけにする場合もあります．参考文献の書式は各学会やその地方の慣習によって異なります．

さらにそれらの文献はあるスタイルに合わせて並べ替えることになります．例えば参考文献を引用した順番で並べ替えるスタイルや，文献の著者名順に並べ替えるスタイルもあります．いずれにしても読者に対しての明確な道しるべとして存在する必要がありますので，その点を考慮した並べ替え方を行います．

例えば参考文献が非常に多い場合，これらを手動で並べ替える作業だけで一晩かかりそうです．これを自動化するために Oren Patashnik 氏が作成した $\text{BibT}_{\text{E}}\text{X}$ というプログラムを使うと便利です．通常は日本語化された $\text{jBibT}_{\text{E}}\text{X}$ を使うことになると思われます．

手動で参考文献を並べ替える場合は thebibliography 環境と呼ばれる専用の環境に `\bibitem` コマンドで文献を追加します． $\text{jBibT}_{\text{E}}\text{X}$ を用いる場合は参考文献データベースである `\file.bib` に文献を追加し， $\text{jBibT}_{\text{E}}\text{X}$ がソーティングを行います．いずれの方法においても本文中では `\cite` で追加した文献を参照します．

4.2 参考文献を手動で並べる場合

まずは文献を手動で並べ替えそれを出力する方法を先に紹介します．参考文献がそれほど多くない場合は文献を手動で並べ替えることが考えられます．そのときは thebibliography 環境を使います．文献を

```
\bibitem[⟨表示形式⟩]{⟨ラベル⟩} ⟨項目⟩
```

のように文書の末尾にまとめます．これらの文献を thebibliography 環境を使って囲みます．

```
\begin{thebibliography}{⟨幅⟩}
\bibitem[⟨表示形式⟩]{⟨ラベル⟩} ⟨項目⟩
\end{thebibliography}
```

参照するときは該当箇所で \cite コマンドを使います．

```
⟨文献⟩~\cite[⟨注記⟩]{⟨ラベル⟩}
```

とします．例を示すと以下ようになります．

```
論文作成をするならば木下是雄による『理科系の作文技術』~\cite{KK1981}は
一読したい著作である．
\begin{thebibliography}{9}
\bibitem{KK1981} 木下是雄．『理科系の作文技術』．中公新書 624．中央公論社，1981．
\bibitem{AY1991} Ada Young．\emph{The Art of awk programming}．
\textbf{5}．Angus Univ．Press．1991．
\end{thebibliography}
```

論文作成をするならば木下是雄による『理科系の作文技術』 [1] は一読したい著作である．

参考文献

- [1] 木下是雄．『理科系の作文技術』．中公新書 624．中央公論社，1981．
- [2] Ada Young．*The Art of awk programming*．5．Angus Univ. Press. 1991.

ここでの thebibliography 環境の引数は '9' となっていますがこれは参考文献の表示形式に割り当てる番号などの最大の幅を指定します．参照している文献が一桁のときは

```
\begin{thebibliography}{9}
```

のようにしますが，文献項目が2桁を超えたときは

```
\begin{thebibliography}{99}
```

と書きます．

▼ 4.2.1 文献の並べ方

thebibliography 環境では文献は自動的に並べ替えられません．そのときは手動で文献を並び替えます．文献の並べ替えの仕方は様々あるのですが書籍の場合

```
\bibitem[⟨表示⟩]{⟨ラベル⟩} ⟨著者⟩. ⟨書名⟩. ⟨シリーズ⟩. ⟨発行年⟩, ⟨出版社⟩. ⟨注記⟩.
```

とするのが一般的です．読者には誰のなんという文献ということが伝わりやすいスタイルです．このように文献を追加し、複数の文献を並べるときは

- 最初の著者の姓をアルファベット順で並べる．
- 同じ著者から複数の文献を参考にしているときは発表年が早い方を先に並べる．

という規則に従います．その他読者に有益だと思える情報があれば、項目の最後に〈注記〉として補足事項を書きます．

例えば 1999 年に未来出版から出版された「未来太郎」の『未来論』という文献があるとします．

```
\bibitem[Mirai 1999]{MT1999} 未来太郎. 『未来論』. 1999, 未来出版.
```

「未来太郎」は ‘Taro Mirai’ という読みになるので、

```
\begin{thebibliography}{Watanabe 2000}
\bibitem[Hokkai 1997]{HM1997a} Michiko Hokkai.
  \emph{Going My Way}. 1997, Future.
\bibitem[Hokkai 1999]{HM1999a} 北海道子.
  それが私の生きる道. 1999, 未来出版.
\bibitem[Watanabe 2000]{NN2000a} 渡辺徹.
  未来大学の見学. 2000, NNN 出版.
\end{thebibliography}
```

のような文献リストがあった場合は、‘[Hokkai 1999]’ と ‘[Watanabe 2000]’ のあいだに入り、次のような出力になります．

```
[Hokkai 1997]    Michiko Hokkai. Going My Way. 1997, Future.
[Hokkai 1999]    北海道子. それが私の生きる道. 1999, 未来出版.
[Mirai 1999]     未来太郎. 『未来論』. 1999, 未来出版.
[Watanabe 2000]  渡辺徹. 未来大学の見学. 2000, NNN 出版.
```

例では北海道子の場合は「北海道子」と ‘Michiko Hokkai’ の 2 通りあります．これは不正確で、同じ著者名の表示は統一します．表示形式は特に指定しなかった場合は昇順に番号付けされます．この表示形式の規則としては「[番号]」とか「[名前 年号]」など作成者と読者に分かりやすいような表示方法にすれば良いでしょう．

しかし、これは自分で文献を並べ替えなどする必要がありますので文献を沢山参照している論文などを作成するときには実用的とは言えません．

4.3 参考文献をプログラムで並べ替えるとき

参考文献が非常に多い場合は手動で並べ替えるのが困難です．参考文献の番号付け，並び替えを行うときに引用順とか発表年順などの書式が存在します． \LaTeX にはこのような手間を省いてくれるプログラムがきちんとあります．日本語化された \LaTeX [14, 24] というのがこれにあたります．原理は簡単で決められたスタイルに合わせて複数の文献を並び替えるだけです．

▼ 4.3.1 \LaTeX の使い方

参考(引用)文献は \LaTeX のソースとは別のファイルに保存します．これを文献データベースと呼びます．ファイル名は任意で良いのですが拡張子は `.bib` となるようにしてください．

▼ 4.3.2 文献データベースの作成

プログラムによって半自動的に文献を並べ替える方法を紹介します．まずは文献データベースと呼ばれるファイルを作ります．名前は `file.bib` ということにしておきます．使い方は一つの文献に対して

```
@<文献の形式>{<ラベル>,  
  <属性 1>={<値 1>},  
  <属性 2>={<値 2>},  
}
```

という記述をします．このような記述を文献の数だけ作成します．参考文献といっても色々ありますので，まずは具体例を見てください．

```
@book{TM2004a,  
  author    = {未来 太郎},  
  yomi      = {Taro Mirai},  
  title     = {未来を深く考える},  
  publisher = {未来出版},  
  year      = {2004},  
  note      = {007//Wa},  
}
```

この文献データベースを記述するための規則があります．

- 一つの文献はアットマーク '@' から始めます．
- '@' の後に 'book' とありますがこれは「文献の形式」を表します．この場合は一般に本屋さんで売っている 'book' であることが分かります．
- 次にその文献の情報を波括弧で括ります．括るときはまずその文献に <ラベル> をつけます．要は目印です．これがないと参照できません．ここでは覚えやすいよう

に ‘TM2004a’ と著者名の頭文字と発行年にしています。

- ‘author’, ‘yomi’, ‘title’, ‘publisher’, ‘year’, ‘note’ などの属性に値を設定します。
- 行末にコンマを記述します。これがないと処理の段階でエラーになります。
- 値は波括弧で囲みます。
- 日本人の著者名は姓名のあいだに半角の空白を入れます。実際に出力されるときは自動的に除かれます。
- 著者名の読み ‘yomi’ には「名」の次に「姓」を書きます。

このような文献データベース file.bib を作成したならば、今度は原稿の本体で、この文献を参照します。参照のコマンドは `\cite` です。方法は 4.2 節の場合と同様です。

▼ 4.3.3 参考文献一覧の出力

一通り参照したら今度は \LaTeX 文書の一番最後に参考文献を出力する記述を追加します。プリアンプルですることはありません。文書の最後のほうで `\bibliography` 命令を使って次のようにします。

```
\bibliographystyle{<スタイル>}
\bibliography{<ファイル名>}
```

〈スタイル〉には文献を並べ替えるスタイルを指定し、〈ファイル名〉には文献データベースの 〈ファイル〉.bib から拡張子.bib を除いた名前を書きます。

これでソースファイルの編集は終わりました。たとえば、ファイルは次のように記述できます。

```
\documentclass{jsarticle}
\begin{document}
この冊子~\cite{TW2004a}を参照してください。
\bibliographystyle{jplain}
\bibliography{ref}
\end{document}
```

しかしこのままでは参考文献の一覧は出力されません。ここで `JBibTeX` というプログラムを使用します。コンソールなどからファイルのある場所に移動して次のコマンドを実行します。

```
$ platex file .tex
$ jbibtex file
$ platex file.tex
$ platex file.tex
```

とすると参考文献が出力されます。

`JBibTeX` を実行すると次のようなメッセージが出力されます。

```
「 This is JBibTeX, Version 0.99c-j0.33 (Web2C 7.5.2)
  The top-level auxiliary file: file.aux
```

```
The style file: jplain.bst
Database file #1: ref.bib
```

上記のメッセージが表示されると、同一フォルダに並べ替え後の文献一覧ファイル `\file\bb1` が生成されます。1 行目には `jBibTeX` のバージョン情報、2 行目には使用した中途ファイル (`file.aux`)、3 行目には文献を出力するスタイル (`jplain.bst`)、最後に文献データベース (`ref.bib`) には何を使ったのかが出力されています。もしも、この段階で何も表示されなければ `jBibTeX` が異常終了したことを意味しますので、`jBibTeX` のログファイル `\file\blg` を参照してください。

参考文献データベースに文献を追加していても本文中で参考していない (`\cite` 命令で参照していない) 場合はその文献は一覧には出力されませんので注意してください。本文中で明示的に参考しなくても文献一覧には出力したいときには `\nocite` コマンドを使います。

```
\nocite{ラベル}
```

`\nocite` は一覧形式のスタイルによって文献の並び替えに影響します。

3 度もタイプセットしなければならないのは面倒かもしれませんが、1 度 `jBibTeX` によって文献一覧 `\file\bb1` を作成しておけば再度文献一覧を作成するのは新しく文献を参照したときだけです。原稿執筆中は特に正式な文献一覧が必要なわけではありませんので、最終的な原稿のタイプセットのときだけ 3 回ほどタイプセットすれば良いことになります。このようなタイプセット処理を半自動的に行うには、`Make` や `latexmk` を使う方法もあります。

▼ 4.3.4 文献の種類及び項目

```
\bibliographystyle{スタイル}
\bibliography{ファイル名, ...}
```

`\bibliographystyle` 命令は参考文献の出力形式を指定します。‘`jplain`’ というのは、昇順に番号付けを行なう一般的な形式です。`\bibliography` 命令で文献データベースを読み込んでいます。これは複数ファイルをカンマで区切って読み込んででもできます。

参考文献としてその文献がどのような形式なのかを指定する必要があります。雑誌の 1 部なのか、論文の 1 部なのかを明示します。

```
@book{label,
```

となっている一行で ‘`book`’ となっている部分に対応する形式を表 4.1 から選んでください。

‘`author`’, ‘`title`’, ‘`publisher`’, ‘`year`’ 以外にも指定することの出来る項目があります。文献リストの各文献に表 4.2 の項目 (フィールド) を追加します。文献の (形式) により必須となる項目が違います。各文献における必須項目と任意項目は表 4.3 の通りです。必須項目は必ず記述しなければならない項目で任意項目は必要に応じて書き足せば良いでしょう。項目のあるなしで文献の並び替えに若干の影響が出ますが、それ程神経質になる

表 4.1 文献の形式

文献の形式	説 明
article	論文誌など発表された論文
book	出版社の明示された本
booklet	印刷，製本されているが出版主体が不明なもの
inbook	書物の一部（章，節，文など何でも）
incollection	それ自身の表題を持つ，本の一部分
inproceedings	会議録中の論文
manual	マニュアル
masterthesis	修士論文
phdthesis	博士論文
misc	他のどれにも当てはまらないときに使う

表 4.2 フィールド名

項目	内容
address	出版社の住所
annote	注釈付きのスタイルで使われる
author	著者名
booktitle	本の名前
chapter	章，節などの番号
crossref	相互参照する文献のデータベースキー
edition	本の版
editor	編集者
howpublished	どのようにして発行されたか
journal	論文誌名
key	著者名がないときに相互引用，ラベル作成などに使われる
month	発行月か書かれた月
note	読者に役立つ付加情報
number	論文誌などの番号
organization	会議を主催した機関名あるいはマニュアルの出版主体
pages	ページ（範囲）
publisher	出版社（者）名
school	論文が書かれた大学
series	シリーズ名
title	表題
volume	論文誌などの巻
year	発行年か書かれた年

表 4.3 文献の種類における必須・任意項目

文献の種類	項目
article	author , title , journal , year
任意	volume , number , pages , month , note
book	author , title , publisher , year
任意	volume , series , address , edition , month , note
booklet	title
任意	author , howpublished , address , month , year , note
inbook	author , title , chapter , pages , publisher , year
任意	volume , series , type , note , address , edition , month
incollection	booktitle , author , title , year publisher ,
任意	editor , volume , series , type , month , note , address , edition
inproceedings	author , title , booktitle , year
任意	editor , volume , series , pages , address , month , organization , publisher , note
manual	title
任意	author , address , edition , month , year , note , organization
masterthesis	author , title , school , year
任意	type , address , month , note
misc	
任意	author , title , howpublished , month , year , note
phdthesis	author , title , school , year
任意	type , address , month , note

必要はありません。

著者 ‘author’ が複数人数のときはカンマで区切るのではなく

```
author={夏目 漱石 and 福沢 諭吉 and 芥川 龍之介}
```

のように ‘and’ を使用します。また著者の苗字と名前のあいだには半角の空白を挿入するようにしてください。‘author’ や ‘editor’ の名前が非常に多いときには名前を

```
author={代表著者 and others}
```

とします。こうすると標準スタイルの jplain では自動的に適切な名前、例えば ‘et al.’ などに置換されます。

▼ 4.3.5 各文献スタイルの出力例

BiBTeX にはどのような文献スタイルが用意されているのかをここで一部分の紹介をします。通常は jplain で問題ないのですが学会によっては参考文献の出力形式を指定される場合があります。使用できるものは欧文の場合、plain, alpha, abbrev, unsrt の 4 つほどで和文の場合は、jplain, jalpha, jabbrv, junsrt となります。他にも WWW 上には個人や学会で文献スタイルを公開していることがありますので、それらを使用することも可能です。

jplain 昇順に通し番号つけるだけの単純なもの。

[1] 野比太郎, 剛太タケル. 2000. 四次元ポケットの考察. NNN 出版.

jalpha 著者が一人の場合は著者は「頭文字 3 文字 年号」で表示し、共著のときは「各著者の頭文字 年号」で表示する。‘key’ 項目を追加することにより表示する頭文字などを変更できる。

[NG 2000] 野比太郎, 剛太タケル. 2000. 四次元ポケットの考察. NNN 出版.

jabbrv 著者名、月、誌名を簡略表記にする。

[1] 野比, 剛太. 2000. 四次元ポケットの考察. NNN 出版.

junsrt 文献を本文中で参照している順番で並べ替える。

[1] 野比太郎, 剛太タケル, 2000. 四次元ポケットの考察. NNN 出版.

▼ 4.3.6 文献の追加例

▷ 例題 4.1 文献データベースに書籍を追加する例です。書籍の出典を明記する場合は、その書籍を特定できる情報を記載することが必須となります。著者名 (author)、書名 (title)、出版社 (publisher)、出版年 (year) の四つは必ず記載します。必要に応じて巻 (volume)、シリーズ (series)、版 (edition) を併記します。

2004 年に技術評論社から出版された奥村晴彦氏の『[改訂第 3 版] LaTeX 2_ε 美文書作成入門』ならば、次のようにします。

```
@book{bibunso,
  author   = {奥村 晴彦},
  yomi     = {Haruhiko Okumura},
  title    = {[改訂第 3 版] \LaTeXe\ 美文書作成入門},
  publisher = {技術評論社},
  year     = {2004},
  note     = {021.49/0k},
}
```

著者名の読み (yomi) は「姓」「名」の順番ではなく、「名」「姓」とします。

▷ 例題 4.2 学会・論文誌などに投稿された論文を追加する場合は、著者名 (author)、題名 (title)、論文誌名 (journal)、発表年 (year) が必須記載項目になります。必要に応

じて論文誌の巻 (volume), 論文誌の番号 (number), ページ番号 (pages) を追加します. 学会誌であれば学会誌の巻や番号がありますので, これも忘れずに追加します. 大沢英一氏による論文誌 *The RoboCup Synthetic Agent Challenge 97* を追加するには次のようにします.

```
@inproceedings{E01997,
  author    = {Eiichi Osawa and others},
  booktitle = {Proceedings of the 15th International Joint Conference
               on Artificial Intelligence: IJCAI-97},
  title     = {The RoboCup Synthetic Agent Challenge~97},
  volume    = 1,
  pages     = {24--29},
  year      = 1997,
}
```

この場合, この論文誌は会議 (conference) 中の論文 (proceeding) ということで inproceedings として分類します. 著者が多くなりすぎる場合は, 代表著者 (姓名の「姓」で並び替えたときに始めるに來る執筆者) だけを書きます. 論文中に代表著者が記されている場合はそれに従います.

例題 4.1 と例題 4.2 の二つの文献を $\text{jBibT}_{\text{E}}\text{X}$ によって処理した結果, 次のような並び替えられた文献一覧 $\langle file \rangle$.bb1 が作成されます.

```
\begin{thebibliography}{9}
\bibitem{H02004} 奥村晴彦.
  \newblock [改訂第3版]  $\text{LaTeX}_{\text{E}}$  美文書作成入門.
  \newblock 技術評論社, 2004.
  \newblock 021.49/Ok.
\bibitem{E01997} Eiichi Osawa, et~al.
  \newblock The robocup synthetic agent challenge~97.
  \newblock In {\em Proceedings of the 15th International Joint Conference
               on Artificial Intelligence: IJCAI-97}, Vol.~1, pp.~24--29, 1997.
\end{thebibliography}
```

この $\langle file \rangle$.bb1 が作成されていれば, 次回のタイプセットで次のような文献一覧が表示されるようになります.

参考文献

- [1] 奥村晴彦. [改訂第3版] LaTeX_{E} 美文書作成入門. 技術評論社, 2004. 021.49/Ok.
- [2] Eiichi Osawa, et al. The robocup synthetic agent challenge 97. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence: IJCAI-97*, Vol. 1, pp. 24–29, 1997.

欧文の文献を参照し, 著者名を代表執筆者のみにした場合は, 慣習的に ‘et al.’ を使います. ページ番号は範囲を示しますので, en-dash ‘–’ を用います.

▼ 4.3.7 文献を同時に複数参照しているとき

複数の文献を同時に参照しているときは‘[3,2,5,1]’となってしまう文献リストの表示が並べ替えられず,‘[1-3,5]’となりません。その場合 Donald Arseneau 氏による cite パッケージを使います。ただし hyperref との併用はできません。このパッケージを利用すれば参考文献が複数ある場合‘[1-3,5]’のように連番をハイフンでつなげ昇順に並べ替えます。プリアンブルで読み込むだけで使用可能です。

▼ 4.3.8 参照の形式を変更する

文献一覧の表示される書式の設定をしたい場合は cite パッケージを使います。このパッケージのオプションとして

nospace 項目のあいだの区切りで単語間空白を挿入しません。

space 項目のあいだの区切りで単語間空白を挿入します。

nosort 並び替えを行いません。

などが用意されています。

```
\usepackage[space]{cite}
```

のように使用してください。設定できるコマンドとして表 4.4 の五つがあります。まずは

表 4.4 cite パッケージで変更できる命令

命令	意味	標準のスタイル
<code>\citeform</code>	個々の項目の修飾	なし
<code>\citepunct</code>	項目の区切り	コンマと小さい空白
<code>\citeleft</code>	リストの左括弧	[
<code>\citeright</code>	リストの右括弧]
<code>\citimid</code>	<code>\cite</code> の任意引数の前に付ける記号	コンマと文字間空白

使用例を見てください。例えば以下のようなファイル mycite.tex を作成します。

```
\documentclass[12pt]{jsarticle}
\usepackage{cite}
\begin{document}
そうです~\cite[p.~130]{First,Second,Third,Sixth,Fifth}.
\begin{thebibliography}{9}
\bibitem{First} First Name. \emph{はじめ}. 1991, 未来出版.
\bibitem{Second} Second Name. \emph{つぎ}. 1992, ある出版.
\bibitem{Third} Third Name. \emph{つぎのつぎ}. 1993, ある社.
\bibitem{Forth} Forth Name. \emph{そのつぎ}. 1995, 未来社.
\bibitem{Fifth} Fifth Name. \emph{さらにつぎ}. 1994, 未来出版.
```

```
\bibitem{Sixth} Sixth Name. \emph{さいご}. 1990, 未来堂.
\end{thebibliography}
\end{document}
```

このまま何も設定しなければ,

そうです [1-3,5,6; p. 130] .

のように並べ替えられ, \cite の任意引数の「p. 130」の前にコンマと小さい空白が挿入されております. さらに項目はコンマで区切られています. 次にこのファイルのプリアンブルに (‘\usepackage’ の後に)

```
\renewcommand\citelmid{; }
\renewcommand\citeleft{ ( }
\renewcommand\citeright{ ) }
\renewcommand\citepunct{,}
```

という記述をしておけば

そうです (1-3,5,6; p. 130).

という出力になります. 個々の項目を修飾するためには \citeform 命令の再定義をします. ローマ数字で番号を表示するときは

```
\renewcommand\citeform[1]{\romannumeral 0#1}
```

とすると

そうです (i-iii,v,vi; p. 130).

のようになります.

4.4 文献一覧の管理

文献の数があまりに多くなると管理するのが大変になります. この場合は GUI インターフェイスで文献データベースを編集することが出来ます. インターフェイスは英語ですが Morten Alver 氏と Nizar Batada 氏が作成した JabRef が

<http://jabref.sourceforge.net/>

にあります.

第 5 章

原稿の出力形式

L^AT_EX の原稿の執筆が終わったらそれを組版（タイプセット）しなければならないのは自明のことですが、どのようなファイル形式にするかは用途により分かれるところです。この章ではどのようなファイル形式があるのか、どうやって変換するのかを説明します。

5.1 出力形式の種類

L^AT_EX の原稿の執筆が終わったらそれを組版（タイプセット）しなければならないのは自明のことですが、どのようなファイル形式にするかは用途により分かれるところです。目的と気分によってその形式を変えます。それぞれの形式がどのような特徴を持っているのかを知っておかなければ、どの形式に変換すれば良いのかが分かりません。ですからまずはどのような形式が存在し、どのような特徴があるのかを紹介します。

DVI DVI は *Device Independent* の略で装置に依存しない汎用のページ記述言語です。画像を含んだり特殊な描画を行っていない原稿の場合はこの DVI ファイルから印刷を行うことができます。装置に依存する命令もこの DVI ファイルの中に記述されており、それを適切に解釈してくれるデバイスドライバがあります。通常はプレビュー作業用に使われています。DVI ファイルは *<file>.dvi* のように拡張子が *.dvi* となります。

PostScript Adobe 社が昔に開発したページ記述言語です。現在のバージョンは 1.3 で Unix 系 OS ではこの PostScript 形式のファイルがプレビュー及び印刷に広く使われています。良く PostScript を省略して PS と書くことがありますし、拡張子は *.ps* になっています。標準ではファイルが圧縮されないのので *<file>.ps.gz* の形で配布されているかもしれません。印刷業界でもこの PostScript 形式が良く使われています。PostScript の仲間に EPS (Encapsulated PostScript) というファイル形式もあります。こちらはベクトル画像などに良く使われています。

PDF PDF は Portable Document Format の略で Adobe 社の開発している PostScript の後継のページ記述言語です。現在のバージョンは 1.5 でプレビューと印刷結果が同程度の品質を得ることができる形式です。世界中で広く使われています。日本語は通りませんが L^AT_EX 形式の原稿を直接 PDF に変換する pdfL^AT_EX というプログラムも存在します。

HTML HTML HyperText Markup Language の略でウェブ上で情報を公開するためのハイパーリンク (Hyper Link) という機能を備えたページ記述言語です。普段ウェブブラウザから見ているページも HTML で記述されています。現在は HTML の後継の XHTML が主流になろうとしています。L^AT_EX と同じようにマークアップ言語です。

以上の形式のほかにもあるのですが、有名な形式はこの四つです。この章ではどのように L^AT_EX の原稿を各形式に変換するかを解説します。

▼ 5.1.1 DVI

DVI とは *DeVice Independent* の略でデバイスに依存しないファイル形式です。通常 L^AT_EX が成形後の結果をまとめるのもこの DVI 形式です。platex などのプログラムで L^AT_EX の原稿を端末などから

```
$ platex filename.tex
```

とすると原稿 $\langle filename \rangle$.tex から正常にタイプセットが終了すれば $\langle filename \rangle$.dvi が生成されます。DVI ファイルにはグラフや画像などの図は挿入されていませんが、それらの情報は DVI ファイルに記載されています。図などの特別な情報を解釈できるかはそのプレビューアやデバイスドライバに依存しています。DVI ファイルはプレビューなどで一時的に組版後の結果を確認するに便利です。Windows では大島利雄氏らが開発している dviout、Unix 系 OS ならば xdvi や pxdvi などが使えます。

▼ 5.1.2 PostScript

机上出版 (DTP) が始まった頃から Adobe 社の PostScript というのが出版業界におけるページ記述言語の標準です。プログラミング言語としての完成度も高く非常に洗練されたページ記述言語です。今でも多くの出版社、印刷所がこの PostScript を採用しています。PostScript は印刷を目的としたファイル形式なのできちんと手順を踏めば高品質な印刷結果を得ることができます。L^AT_EX もこの PostScript 形式への出力が可能となっています。この PostScript 形式のファイルは Ghostscript と呼ばれるプログラムを使うことにより、コンピュータ上で閲覧したり、プリンターで印刷することができます。

▼ 5.1.3 PDF

Adobe 社が開発した電子文書形式で PDF という形式があります。PDF は *Portable Document Format* の略で、パソコンの画面からでも印刷したのと寸分違わぬ表示を得ることができます。マニュアルの配布や資料の配布ではこの PDF 形式が広く用いられています。

5.2 DVI を PS に dvipsk

Tomas Rokicki 氏が作成し、Karl Berry 氏が Kpathsearch に対応させた dvipsk を使うと DVI ファイルを PS ファイルに変換できます。dvipsk というプログラムは Windows の方は dvipsk、Unix 系 OS の方は dvips という名前が付いていると思います。Red Hat の場合は pdvips という名前になっています。使い方は端末などから

```
$ dvips <オプション> <引数> filename.dvi
```

とするだけです。拡張子 .dvi は省略しても構いません。この dvipsk を実行するときのコマンドラインオプションが多数あります。主なオプションを載せておきます。

- D <解像度> 出力する解像度を dpi 単位で指定します。
- o <ファイル名> 出力するファイル名を指定します。
- t <サイズ> a0 から a8、b0 から b8 の範囲で用紙の大きさを指定します。標準では用紙が欧文標準の letter サイズになっている場合もあります。
- T <横幅>,<高さ> 用紙の大きさを単位付き直接指定します。‘21cm,27cm’ のように使います。このようにしなくとも原稿のプリアンブルで


```
\AtBeginDvi{\special{papersize=210mm,270mm}}
```

 としても同じことになります。
- p <ページ番号> 出力する最初のページを指定します。ただし L^AT_EX の原稿中のページ番号を参照します。
- l <ページ番号> 出力する最終のページを指定します。ただし L^AT_EX の原稿中のページ番号を参照します。
- pp <ページリスト> 出力するページ範囲を指定します。これも L^AT_EX のページ番号に依存します。11,21-35 のようにコマンドで複数ページ指定することもできます。
- P <設定> 設定ファイルを読み込みます。標準では config.ps というファイルを読み込みます。

標準ではビットマップフォントが埋め込まれるようになっていいますので、config.pdf を読み込むために

```
$ dvipsk -Ppdf filename
```

 としてください。このようにすると Type1 形式のアウトラインフォントが埋め込まれるようになります。

複数ページからなる DVI ファイルから特定のページだけを EPS 形式にしたいというならば

```
$ dvipsk -E -Pdl -pp14 -o outp14.eps input
```

とします。このようにして抽出した EPS 形式のファイル outp14.eps は EPS 画像として再利用できます。

5.3 DVI を PDF に Dvipdfmx

Mark Wicks 氏が作成した Dvipdfm [28] を使うと DVI ファイルから PDF を作成できます。現在 Dvipdfm は平田俊作氏と Cho Jin-Hwan 氏が中心となって活動している Dvipdfmx project team によってさらに改良が加えられ Dvipdfmx へと改名しています。Dvipdfm は少々古くなっていますので後継の Dvipdfmx を使うことをお勧めします。

Dvipdfmx の主な機能は PDF ブックマーク, HyperTeX, TPIC スペシャルなどをサポートしています。画像ファイルは JPEG, PNG, EPS, EPDF ファイルのバウンディングボックスという, 画像のサイズ情報があればそのまま PDF に取り込むことができるようになります。

Dvipdfmx にはコマンドラインオプションによってある程度の出力結果の設定を行います。主要なオプションは以下の通りです。

- o 〈ファイル〉 出力するファイル名を指定します。標準では 〈file〉.dvi を指定すれば 〈file〉.pdf が作成されます。
- p 〈サイズ〉. 出力する用紙のサイズを指定します。標準では a4。指定できるサイズは letter, a6, a5, a4, a3, b5, b5, b4, b3, b5var などです。
- s 〈範囲〉 出力するページの範囲を指定します。ハイフンを使うと範囲を指定, コマを使うと複数の範囲を指定できます。例えば ‘-s 3-5,10-20’ とすると 3-5 ページと 10-20 が一つの PDF に出力されます。ハイフンの片方に何もないとそれ以前か, それ以降のページを全て含みます。‘-s 15-’ とすると 15 ページ以降全てを出力します。他にもページを逆順にすることもできます。
- r 〈解像度〉 PDF ファイルの解像度を指定します。標準は 600dpi になっています。
- V 〈バージョン〉 PDF のバージョンを指定できます。2 から 5 までのバージョンを指定できますが, 古いバージョンを指定すると意図しない結果になることがあります。互換性を優先しなければならないときなどに使います。
- z 〈数字〉 圧縮率を指定します。圧縮率は 0-9 まで指定でき 9 が最高です。標準は 9 です。ビットマップ画像などの画質を落としたい場合は 0 などにとすると良いでしょう。
- v 処理内容を標準出力に詳しく表示します。

例えば DVI ファイルの 15 ページから 20 ページを PDF に変換したいときは

```
$ dvipdfmx -s 15-20 -o output.pdf input.dvi
```

のようにします。入力ファイルの拡張子 .dvi は省略しても構いません。

PDF ファイルを Adobe Reader や Acrobat Reader などで閲覧しているときに dvipdfm による DVI ファイルの変換を行うと **** ERROR ** Unable to open output.pdf** というメッセージを表示してエラーになります。1 度開いている PDF ファイルを閉じてから, 再度変換するようにします。

5.4 画像ファイルの扱い

一般的な図の扱い方については第 8.3 章を参照してください。L^AT_EX は JPG, PNG などのビットマップ画像や、曲線の描画などの多くをデバイスドライバと呼ばれる外部の装置に依存しています。そのため、L^AT_EX で画像ファイルを扱う場合は、まずデバイスドライバを用途別に選択することになります。

L^AT_EX で画像を張り込む場合は graphicx パッケージを使うことになります。dvipsk を使っている場合はパッケージオプションを次のように dvipsk ではなく、dvips とします。

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

Dvipdfmx を使っている場合は、dvipdfmx ではなく、dvipdfm とします。

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

dvipsk は基本的に EPS 画像の張り込みにしか対応していませんので、スクリーンキャプチャなどのビットマップ画像は ImageMagick などを使って EPS に変換しておきます。

dvipdfm において JPEG, PNG, PDF, EPS などの画像ファイルはバウンディングボックスという画像の(原点座標を含む)サイズ情報があれば張り込むことが可能です。dvipdfm 付属の ebb というプログラムで画像のバウンディングボックスの作成をすれば、JPEG, PNG, PDF, EPS を直接 PDF に張り込めます。具体的な手順としてはファイルのあるディレクトリで

```
$ ebb filename.jpg
```

とすれば拡張子が.bb の *filename*.bb というファイルが作成されます。作成された *filename*.bb を見てみると

```
%%Title: ./filename.jpg
%%Creator: ebb Version 0.5.2
%%BoundingBox: 0 0 595 842
%%CreationDate: Tue Dec 30 13:04:10 2003
```

のように〈ファイル名〉, 〈作成プログラム〉, 〈バウンディングボックス〉, 〈作成日時〉の情報が出力されます。沢山 *filename*.bb のファイルを保存しておくのが好ましくない場合は、該当する画像ファイルを読み込んでいる箇所では、

```
\includegraphics[bb={0 0 595 842}]{filename.jpg}
```

とすれば *filename*.bb がなくても良いことになります。使用する画像のファイル名の〈ファイル名〉. 拡張子は 'filename.png' のように 〈8 文字〉.3 文字したほうが互換性の上で安全です。

Dvipdfmx の場合は基本的に PDF, JPEG, PNG, MetaPost 形式の画像しかサポートしておりませんので、EPS 形式の画像は何らかの形で PDF に変換してから取り込むことになります。この EPS ファイルは Ghostscript の 'pdfwrite' というデバイスを

使って変換することがほとんどです．そのときに `epstopdf` か `ps2pdf14` などを使います．`epstopdf` は PDF に EPS の BoundingBox を反映してくれます．`ps2pdf` 系を使う場合は PDF に BoundingBox がうまく反映されないので以下のようなシェルスクリプト `eps2pdfs`

```
#!/bin/bash
EPS='ls *.eps';
for fig in $EPS; do
    epstopdf $fig
    $f='basename $fig .eps'
    grep "%BoundingBox:" $fig > $f.bb
done
```

を作成し PATH の通っている場所 (`/usr/local/bin/` など) に複製したならば

```
$ ./eps2pdfs
```

とすると同ディレクトリの EPS ファイルが全て PDF に変換されます．`<file>.eps` があったとすればこれは `<file>.pdf` と `<file>.bb` が作成されます．このようにして EPS から PDF に変換したファイルは \LaTeX の原稿で次のように取り込むことができます (行頭のパーセントは取り除いてください) ．

```
%\documentclass{jarticle}
%\usepackage[dvipdfm]{graphicx}
%\begin{document}
\includegraphics[width=3.5cm]
{images/ridt91}
%\end{document}
```



`dvipsk` と `Dvipdfmx` の両方を使っているという (Unix 系 OS の方で普段は PostScript で印刷していて、提出用に PDF を作成するなど) 場合は `images` ディレクトリを作成し、そこに `<image>.eps`、`<image>.pdf`、`<images>.bb` の三つのファイルを置きます．次に原稿中で次のように `\includegraphics` 命令を使うとき拡張子を省略します．

```
\includegraphics[width=3cm]{images/ridt91}
```

すると `graphicx` パッケージに渡されたパッケージオプションに従って、張り込まれる画像の優先順位が変わりますので、`dvips` を指定している場合は EPS が、`dvipdfm` を指定している場合は PDF が張り込まれるようになります．次のように `graphicx` の読み込みの仕方を変更するだけです．

```
%\usepackage[dvips]{graphicx} % dvipsk の場合
%\usepackage[dvipdfm]{graphicx} % Dvipdfmx の場合
```

第 6 章

コマンドとマークアップ

マークアップ言語とは何なのか、マークアップで何が実現できるのか、それを \LaTeX でどのように実現するのかという基本的な部分を紹介します。

6.1 マークアップ言語とは？

数十年前に、文書に対して入れ子型の論理構造を与えることによって汎用性を持たせ、人間が直接理解できる文書の記述に関して研究がなされたそうです。その中でもウェブページを記述する言語として HTML: Hyper Text Markup Language というものが提案されました。現在は XHTML: Extended Hyper Text Markup Language へと進化し、統一化が図られています。 \LaTeX も HTML や XHTML と同じようにマークアップ方式を採用しているページ記述言語です。

6.2 記号とコマンド

\LaTeX はコンピュータプログラムですから、人間の意図を知るためには何か特別な命令を人間から受け付けることになります。そのため原稿にはコマンドと呼ばれる特別な記号の綴りを使ったり、いくつかの記号に特別な意味を持たせます。

▼ 6.2.1 記号の分類

\LaTeX ではユーザが出力したい意味を理解するために全ての記号に \LaTeX なりの意味を割り当てています。人間がキーボードから ‘<’ という記号を入力しても数学の比較演算子とは知ることができません。‘\$<\$’ としなければ「ここからここは数式であり、‘<’ は比較演算子として使う。」という意味を理解してくれません。そのため \LaTeX に入力を与えるユーザは \LaTeX の文法を覚える必要があります。詳しく覚える必要はありませんが

`\ { } $ & # ^ _ ~ %`

という 10 個の記号には特別な意味があることを覚えてください。

▼ 6.2.2 コマンド

テキストを入力していると‘<’というキーボードからの入力が‘\’になってしまいます。これは一体どういうことでしょうか。考えてみると‘<’という入力は‘<’という記号を出力するという命令ではなく別の命令、‘\’を出力するという命令に割り当てられていると考えられます。さらに\%のようなバックスラッシュ（円）の次に記号が来るようなコマンドも存在します。ここで \LaTeX のコマンドは「バックスラッシュと文字列」という話ではないことが分かります。正確には「バックスラッシュと記号の綴り」をコントロールシーケンスと呼び、特殊記号1文字をコントロールシンボルと呼びます。 \LaTeX におけるコマンドは大きく分けると三つに分類できます。

コントロールシーケンス バックスラッシュ‘\’（‘¥’）と記号の綴り。制御綴りと訳されることもあります。これを本書では狭義のコマンドとして表現しています。

コントロールワード バックスラッシュと英字の綴り。例えば‘\section’など。

コントロールシンボル バックスラッシュと英文字以外の綴り。例えば‘\3’とか‘\#’など。

コントロールスペース バックスラッシュとスペース一つの綴り。‘\ ’のこと。
特殊記号 特別な意味を持つ記号。予約文字と呼ばれることもあります。例として‘{’、‘\$’など。

英数字など バックスラッシュの付かない普通の文字列。

現段階では大きく分けると

- バックスラッシュと文字列の綴り。
- 特殊な記号。
- 普通の文字列。

の三つがあることを理解してください。本冊子では制御綴り（コントロールシーケンス）のことをコマンドと呼び命令、宣言、環境の三つに分類します。

命令 特定の処理がそのときに実行されるコマンド。他の参考書ではこの命令のことをコマンドと呼ぶことが多いようです。引数を取ることがあり、その引数のことを要素と呼んだり、オプションと呼んだりします。例として\maketitleや\sectionなどがあります。

宣言 特定の処理がそれ以降継続して行われるコマンド。処理の適用される範囲を限定する（グルーピング）こともできる。引数をとることは稀。よく宣言のことも命令や宣言方命令とか宣言型コマンドと呼ばれます。例として\ttfamilyがあります。宣言型のコマンドは命令に比べると少ないので、本冊子でも断り書きとして宣言型コマンドと呼ぶことが多いです。

環境 \begin{何々}と\end{何々}によって要素を囲むコマンド、または囲まれている領域のこと。引数を取ることがあります。例としてdocument環境などがあり

ます。

▼ 6.2.3 コマンドの定義

L^AT_EX の原稿では新しい命令などの定義をすることができます。

```
\newcommand{<命令>}[<整数>][<標準値>]{<定義>}
\renewcommand{<命令>}[<整数>][<標準値>]{<定義>}
```

`\newcommand` についてですが、この命令によって、まだ定義されていない `<命令>` を新規に定義することができます。

```
\newcommand{\example}{これは例です。}
```

として、本文中で `\example` と記述すると

これは例です。

という出力になります。さらに

```
\newcommand{\example}[2]{#1 は#2 です。}
```

として、本文中で `\example{ボブ}{背が高い}` と記述すると、

ボブは背が高いです。

という出力になります。この `\example` 命令に任意引数があっても良いことを宣言するためには次のようにしますが、任意引数も引数の総和に勘定します。

```
\newcommand{\example}[2][未来]{%
  私は#1#2 にいます。}
\example{大学} \example{出版}\par
\example[] {大学} \example[函館]{出版}
```

私は未来大学にいます。 私は未来出版にいます。
私は大学にいます。 私は函館出版にいます。

このように任意引数や必須引数の定義なども、`\newcommand` 命令を使うことにより実現できます。定義の中で引数は `'#<n>'` として扱い、1 から 9 までの整数が使えます。このような定義は数式の記述などに威力を発揮します。

```
\newcommand{\seq}[2][n]{%
  \{#2_{0},#2_{1},\ldots,#2_{#1}\}}
数式の集合もマクロを使って $\seq{a}$ や
\seq{k}{x}$ とできます。
```

数式の集合もマクロを使って $\{a_0, a_1, \dots, a_n\}$ や $\{x_0, x_1, \dots, x_k\}$ とできます。

`\newcommand` では任意引数をつつしか設けることができませんが、引数は合計 9 個まで使うことができます。`\renewcommand` では一度定義した命令を再度定義することができます。

さらに通常 L^AT_EX でよく見かける環境型のコマンドの定義に関しては以下の四つの命令が使えます。

```
\newenvironment{<命令>}[<整数>][<標準>]{<始め>}{<終わり>}
\renewenvironment{<命令>}[<整数>][<標準>]{<始め>}{<終わり>}
```

`\newenvironment` では環境の始めの部分と終わりの部分を定義して、新たに環境型の命令を作成します。引数に関する扱いは `\newcommand` と同じです。`\renewenvironment` については一度定義した環境型のコマンドを再度定義する機能があります。中央揃えして書体を強調したい環境は次のように `cemph` のように作成します。

```
\newenvironment{cemph}%
  {\begin{center}\begin{em}}%
  {\end{em}\end{center}}
ここの文章は通常通り出力され、
                                     この文章は中央揃えで強調表示
\begin{cemph}
ここの文章は中央揃えで強調表示
                                     されましたか？
\end{cemph}
ここの文章は中央揃えで強調表示
                                     されましたか？
```

▼ 6.2.4 文字やコマンドの区切り

私たち人間はある文や節の区切りをどのように判断しているのでしょうか。一つは文と文のあいだや単語と単語のあいだに挿入される空白です。空白は文字列の区切りを示し、その空白には意味の区切りがあります。では節はどうでしょうか。一つの例としてメールアドレスの場合を考えてみます。メールアドレスはそもそもコンピュータ上で手紙のやり取りをするための住所ですからコンピュータが分かりやすい表現になっていますが、人間にも分かりやすい表記になっています。仮に

```
name@server.co.jp
```

というメールアドレスがあったとします。するとこれは

```
'name' '@' 'server' '.' 'co' '.' 'jp'
```

に分けられます。それぞれ

name メールアドレスを使っている人の「名前」。

@ '@' は 'at' の意味でもあって、これ以降の文字は「住所」を表すことを示す。

jp その人の「国」を表す。

co その人がどんな「地域（組織）」に所属しているのかを表す。

server 地域の中のどこにいるのかをあらわす住所。

. 住所を区切るために使われている。

という意味合いを持っています。住所の区切りが空白ではなくピリオドなのは仕方のないことです。コンピュータの世界ではなるべく文字列は空白を含んでいないほうが処理が行いやすいのです。さて、これはどのようにして区切りを見つけたのでしょうか。メールアドレスの例では '@' や '.' を文字の区切りとして住所を判定しています。L^AT_EX でも同じ

ようなことをやっています。

このことから \LaTeX においての命令の定義には英字のみにすることが求められるようです。そして英字以外の文字列は、そこをコマンドの区切りとして英字以外の文字列を引数として受け取るということです。

この文字の分類を利用して \LaTeX ではマクロの中において特別な処理をしています。マクロは容易に変更してもらっては困るのでユーザからそのマクロを簡単に変更されないようにしています。その方法の一つとしてマクロの中ではアットマーク '@' を英字と同じ分類として扱うのです。'@' を英字と同じ分類にすると、そこでコマンドは区切られないので

```
\newcommand{\hoge}[2]{あ, #1 だよ, ほら#2}
```

のような定義ができるわけです。そして

```
\newcommand{\hoge}{\hoge}
```

という定義がマクロの中では可能なので、ユーザーから `\hoge` 命令の実態を隠すことができます。

実際ヘッダーやフッターを自分流にカスタマイズしたいときはそれらの命令に '@' が含まれているために変更できない、という事態に陥ります。マクロで行っていること、'@' を英字と同じ分類にしてコマンドを定義するためには

```
\makeatletter ('@' を英字と同じ分類にする.)
\makeatother ('@' を違う分類にする.)
```

という二つの命令を使います。この命令の中身を見てみると

```
\def\makeatletter{\catcode'\@11\relax}
\def\makeatother{\catcode'\@12\relax}
```

となっています。どうやら '@' の `\catcode` というものを 11 にすると英字と同じになり、12 にすると違う分類になるようです。このような記号の分類を通常カテゴリーコードと呼びます (表 6.1 参照)。

そのため何かマクロの中のコマンドに変更を加えたいときは

```
\documentclass{jsarticle}
\makeatletter
\newcommand{\hoge}[2]{あ, #1 だよ, ほら#2}
\newcommand{\hoge}{\hoge}
\makeatother
\begin{document}
\hoge{函館}{未来}.
\end{document}
```

のように '@' を含む箇所を `\makeatletter` と `\makeatother` で囲んであげます。

表 6.1 カテゴリーコードの一覧

カテゴリ	意味	標準での割り当て
0	エスケープ文字	\ (¥)
1	グループの開始	{
2	グループの終わり	}
3	数式モードの制御	\$
4	配列の要素の区切り	&
5	行末文字	〈改行〉 (0x0D)
6	パラメータ文字	#
7	上付き文字	^
8	下付き文字	_
9	無視される文字	なし *1
10	空白	□
11	英文字	A…Z と a…z
12	そのほかの文字	(! ? 1 2 @ など
13	アクティブ文字	~
14	コメント文字	%
15	無効文字	〈デリート〉 (0x7E)
以下三つは日本語 T _E X のもの		
16	第 1・第 2 水準の漢字	亜, 井 など
17	かな, 全角アルファベット	あ, ア, a, A など
18	その他の全角記号	, 【 など

*1 標準では割り当てられていない

▼ 6.2.5 コマンドの引数

引数と取るコマンドに対して文字列を渡した場合の挙動は予想しやすいと思います。ではコマンドに対して制御綴りを渡した場合はどうなるでしょうか。

```
\newcommand{\twoarg}[2]{#1! #2? }
\twoarg a b とか \twoarg{はこだて}{未来}
とか, さらに \twoarg{\LaTeX}{\LaTeXe}
```

a! b? とかはこだて! 未来? とか, さらに L^AT_EX!
L^AT_EX 2_ε?

どうやら引数を取るコマンドに対してさらに制御綴りを引数に与えても良いようです。では次の場合はどうでしょうか。

```
\newcommand{\twoarg}[2]{#1! #2? }
\twoarg{\LaTeX}\LaTeXe
\twoarg{\LaTeX2}\LaTeX3
```

L^AT_EX! L^AT_EX 2_ε? L^AT_EX! 2? L^AT_EX3

これには 6.2.4 節でやったことが含まれています。‘L^AT_EX’ と ‘2’ のあいだで語が区切

られて解釈されているので二つ目の引数に ‘2’ だけが渡されています。

6.3 グルーピング・入れ子構造

T_EX/L^AT_EX では変数のスコープ（有効範囲）という概念を持っています。

まず変数には「限られた範囲だけ有効」な局所変数と「全ての範囲で」有効な大域変数の 2 通りがあります。L^AT_EX においてもこれは重要な話で、この有効範囲（スコープ）を決めるのが波括弧です。

書体変更の宣言でどのように書体が変更されるのかを見てみましょう。今回はファミリーを変える `\ttfamily` とシェープを変える `\itshape`、そして普通の書体に戻す `\normalfont` という三つの宣言型コマンドを使います。

```
roman {\ttfamily tt {\itshape it} tt          roman tt it tt it roman
\normalfont it} roman
```

ここでややと気づいていただきたいのは `\ttfamily` という宣言が二つの括弧の中にも影響しているという点です。先ほどの変数の代入ではこのようにはなりません。どうやら書体の宣言は、その宣言をした場所から内側の括弧までもが有効範囲になっているようです。これは現在の L^AT_EX の仕様です。宣言ではなく命令としても結果は同じになります。

```
roman \texttt{ tt \textit{it} tt          roman tt it tt it roman
\normalfont it} roman
```

しかし `\normalfont` 命令を使うとタイプライタ体の有効範囲でもそこで通常の書体に戻ってしまいます。こう考えると影響を与えたくない括弧の内側の領域には `\normalfont` を使うと良いことになります。

```
roman {\ttfamily tt {\normalfont
  \itshape it} tt} roman\par          roman tt it tt roman
roman \texttt{tt {\normalfont      roman tt it tt roman
  \textit{it}} tt} roman
```

命令ではなく宣言型のコマンドのいくつかは括弧の内側まで影響するので、その属性を受けないようにするための工夫が必要になります。

6.4 宣言と命令の違い

例えば `center` 環境のコマンドを考えると、なぜ環境の内側では全ての行が中央揃えになるのでしょうか。一つは

```
\begin{center}
```

によってグループが始まり、

```
\end{center}
```

によってグループが終わらせているために、どこからどこまでが中央揃えなのかが分かっているでしょう。「これをまさに中央揃えにしてください。」と言うよりは「ここからここまでを中央揃えにしてください。」というコマンドのほうが都合が良いことに気づくでしょう。非常に長い文章の場合は `\centering` 命令を使い

```
{\centering まさにここが中央揃えにされます。}
```

とするよりも `center` 環境として

```
\begin{center}
ここから終わりまで中央揃えになります。
\end{center}
```

としたほうが分かりやすいでしょう。そう考えるとコマンドには

宣言型コマンド 使用してからそれ以降ずっと有効なコマンド。環境型のコマンドに使われたり、単独で使われる。

命令型コマンド 使用した場所では有効なコマンド。通常は引数に与えられたものを処理する。

の二つがあることになります。

例として命令型の `\textsf` と宣言型の `\sffamily` を考えてみましょう。命令型の場合は

```
Roman. \textsf{Roman?}\par This is sans serif family.} Roman!
```

のような使い方はできませんが、宣言型ならば新規に `sffont` 環境を定義できます。

```
\newenvironment{sffont}{\sffamily}{}
Roman.
\begin{sffont}
Roman? \par This is sans serif family.
\end{sffont}
Roman!
```

Roman. Roman?
This is sans serif family. Roman!

宣言型のコマンドはそれ以降ずっと有効なので有効範囲を決めてあげます。`\sffamily` などの書体を変更するコマンドはグルーピングする必要があります。

```
Roman! {\sffamily sans serif family.}
Roman! Roman! sans serif family. Roman!
```

今まで使ってきた `\begin{<何々>}` と `\end{<何々>}` というコマンドは、このグルーピングの作業をやっているのです。補足的なことですが

```
\begin{<何々> <要素> \end{<何々>}
```

というのは \LaTeX の中で

```
{<何々> <始めの処理> <要素> <終わりの処理>}
```

に変換されるので `\sffamily` のような宣言も

```
Roman?
\begin{sffamily}
This is sans serif family.           Roman? This is sans serif family. Roman!
\end{sffamily}
Roman!
```

とできます。こうすると特に長い文章が読みやすくなります。

6.5 相互参照

文章の論理構造を明確にしてくれるものの一つに相互参照があります。相互参照の仕方は参照したいものにラベルを貼り、挿入したい場所でラベルを参照するという二つの作業に分けられます。相互参照できる項目は以下の四つ程に限られています。

- 章節命令 (`\section` 命令など)
- 番号付き数式 (`equation` 環境など)
- `float` 環境の要素 (図や表など)
- `enumerate` 環境内の個々の項目

要は通し番号のついているものには付けても良いようです。ラベルは単純に貼りたいものに `\label` 命令で

```
<参照したい要素>\label{<ラベル名>}
```

のようにします。参照の仕方にはその番号を参照する `\ref` とページを参照する `\pageref` の 2 通りがあります。

```
\ref{<ラベル>} (通し番号)
\pageref{<ラベル>} (ページ番号)
```

参照の仕方は以下ようになります。通し番号を参照する `\ref` 命令は `\section` 命令のようなものを参照するときに非常に便利です。

```
%\section{相互参照}\label{sec:xr}
詳しくは\pageref{sec:xr}~ページの
\ref{sec:xr}~節で述べているのでそ
ちらを参照されたい。
```

詳しくは 73 ページの 6.5 節で述べているのでそちらを参照されたい。

相互参照や目次を作成しているときはタイプセットを 3 回程行う必要があります。ラベルの名前が重複しないように工夫することも必要です。

▼ 6.5.1 相互参照の仕組み

節 (見出し) や図表には通し番号付けます。これは同じ名前の節 (見出し) が同じページに存在しても区別できるという利点があります。そして節 (見出し) を参照するときはその番号を示します。このような機能を実現するために \LaTeX ではカウンタを使います。

ユーザーが特にこのことを意識しなくても半自動的に番号付けなどをやってくれます．一応さわり程度にはその仕組みを説明します．

相互参照する対象が通し番号ですので，節なら節などの要素に応じたカウンタがあらかじめ用意されています． \LaTeX では表 6.2 の通りにあらかじめ定義されているカウンタがあります．カウンタは「素の番号」と実際に出力すべき「表示用の番号」と「参照用の文

表 6.2 あらかじめ定義されているカウンタ名

カウンタ名	割り当て
part	部見出し
chapter	章見出し
section	節見出し
subsection	小節見出し
subsubsection	小小節見出し
paragraph	段落見出し
subparagraph	小段落見出し
page	ページ番号
equation	式番号
figure	図見出し
table	表見出し
footnote	脚注番号
mpfootnote	minipage 環境中の脚注番号
enumi	一つ目の階層の enumerate 環境の番号
enumii	二つ目の階層の enumerate 環境の番号
enumiii	三つ目の階層の enumerate 環境の番号
enumiv	四つ目の階層の enumerate 環境の番号

字列」の三つの要素を持っています．

▼ 6.5.2 カウンタ

章見出しやページには通し番号が振られています．これらは \LaTeX カウンタによって制御されています．カウンタはプログラミング言語で言えば `int` 型（整数）の変数です．カウンタ変数の仕組みや制御の方法を少しは知っておいたほうが後々便利です．この章では変数の基礎を説明します．

例えば `jsbook` クラスで章 (`\chapter`) の下の階層の節 (`\section`) 用のカウンタを定義するには

```
\newcounter{section}[chapter]
```

とします．このようなカウンタの定義には次の命令が使えます．

```

\newcounter{<カウンタ名>}[<親カウンタ名>]
\setcounter{<カウンタ名>}{<数値>}
\addtocounter{<カウンタ名>}{<数値>}
\stepcounter{<カウンタ名>}
\refstepcounter{<カウンタ名>}
\value{<カウンタ名>}

```

`\newcounter` でカウンタを新設します。`\setcounter` は数値を代入し、`\addtocounter` は数値を足し、`\stepcounter` はカウンタの値を一つだけ増やします。`\refstepcounter` はカウンタを後から参照できるようにラベル用が用意されます。`\stepcounter` と `\refstepcounter` によって親カウンタが増えるとその子であるカウンタは 0 にリセットされます。`\value` はカウンタから親カウンタの値や文字列などを取り除いた純粋なカウンタの値が得られるコマンドです。

カウンタの表示形式は変更するものに以下があります。

```

\arabic{<カウンタ名>}      ( 1, 2, 3, ... )
\roman{<カウンタ名>}      ( i, ii, iii, ... )
\Roman{<カウンタ名>}      ( I, II, III, ... )
\alph{<カウンタ名>}      ( a, b, c, ..., z )
\Alph{<カウンタ名>}      ( A, B, C, ..., Z )
\fnsymbol{<カウンタ名>}   ( *, †, ‡, ... )

```

例えば節 (`\section`) の見出し番号をローマ数字に変更するのであれば、節見出し用のカウンタ `'section'` を次のように再定義します。

```
\renewcommand{\thesection}{\Roman{section}}
```

6.6 相互参照の工夫

例えば色について考察した章の中に同じような節見出し、表、図などが存在していたとしましょう。それらのラベルは重複してはいけませんので、何らかの工夫をしておいたほうが得策です。良く使われている方法に表 6.3 のように要素に応じてラベルに対して接頭語を付けます。簡単な例として節見出しを参照するときは

表 6.3 要素に応じたラベルの貼り方

要素	接頭語	対象
章見出し	chap:	<code>\chapter</code>
節見出し	sec:	<code>\section</code>
図	fig:	figure 環境中の <code>\caption</code> 命令
表	tab:	table 環境中の <code>\caption</code> 命令
式	equ:	番号付きの数式 (<code>\equation</code> 命令や <code>eqnarray</code> 環境)

```
\section{加法混色}\label{sec:addmix}
ほげは、ほげほげ。
\section{減法混色}\label{sec:submix}
\ref{sec:addmix}~節\pp{\pageref{sec:addmix}ページ}では、ほげほげ。
```

という入力になります。

これは表 6.3 の規則にしたがって何のマクロも作成せずに手動でやるとちょっと大変なことになります。

```
\section{加法混色}\label{sec:addmixcolor}
点 $i$ における色 $c_i$ は式~\ref{equ:addmixcolor}によって決まる。
\begin{equation}
c_i = r_i + g_i + b_i\label{equ:addmixcolor}
\end{equation}
その関係は表~\ref{tab:addmixcolor}となる。
\begin{table}[htbp]
% ここに表が入る。
\caption{加法混色の表}\label{tab:addmixcolor}
\end{table}
またそれらを図式すると図~\ref{fig:addmixcolor}となる。
\begin{figure}[htbp]
% ここに図が入る。
\caption{加法混色の図}\label{fig:addmixcolor}
\end{figure}
\section{減法混色}\label{sec:submixcolor}
\ref{sec:addmixcolor}~節 (\pageref{sec:addmixcolor}~ページ) ではほげ。
```

3.1 加法混色

点 i における色 c_i は式 3.1 によって決まる。

$$c_i = r_i + g_i + b_i \quad (3.1)$$

その関係は表 3.1 となる。

表 3.1 加法混色の表

またそれらを図式すると図 3.1 となる。

図 3.1 加法混色の図

3.2 減法混色

3.1 節 (5 ページ) ではほげ。

表 6.3 のような規則に従いマクロを作ります。マクロ側で自動的に接頭語を付けてくれれば人間の作業が減りますし、ミスも少なくなります。

```
\newcommand*{\chaplab}[1]{\label{chap:#1}}% 章のラベル
```



```

\newcommand*{\chapref}[1]{第~\ref{chap:#1}~章}% 章の参照
\newcommand*{\seclab}[1]{\label{sec:#1}}% 節のラベル
\newcommand*{\secref}[1]{\ref{sec:#1}~節}% 節の参照
\newcommand*{\figlab}[1]{\label{fig:#1}}% 図のラベル
\newcommand*{\figref}[1]{\ref{fig:#1}}% 図の参照
\newcommand*{\tablab}[1]{\label{tab:#1}}% 表のラベル
\newcommand*{\tabref}[1]{表~\ref{tab:#1}}% 表の参照
\newcommand*{\equlab}[1]{\label{equ:#1}}% 式のラベル
\newcommand*{\equiref}[1]{式~\ref{equ:#1}}% 式の参照

```

このようなマクロを作成しておけば先程の入力は幾分簡略化できるでしょう。

```

\section{加法混色}\seclab{addmixcolor}
点iにおける色ciは\eqref{addmixcolor}によって決まる。
\begin{equation}
c_i = r_i + g_i + b_i\eqlab{addmixcolor}
\end{equation}
その関係は\tabref{addmixcolor}となる。
\begin{table}[htbp]
% ここに表が入る。
\caption{加法混色の表}\tablab{addmixcolor}
\end{table}
またそれらを図式すると\figref{addmixcolor}となる。
\begin{figure}[htbp]
% ここに図が入る。
\caption{加法混色の図}\figlab{addmixcolor}
\end{figure}
\section{減法混色}\seclab{submixcolor}
\secref{addmixcolor}(\pageref{sec:addmixcolor}~ページ)ではほげ。

```

さて、最後の1行を見てみると

```

\secref{addmixcolor}(\pageref{sec:addmixcolor}~ページ)ではほげ。

```

という記述が見受けられます。これは人間が手動で接頭語 sec: を付けなければならない例です。これもミスを誘い出す一因になるかもしれませんのでページ番号も参照するようなマクロを作ります。

```

\newcommand*{\fullchapref}[1]{第~\ref{chap:#1}~章 (\pageref{chap:#1}~ページ)}
\newcommand*{\fullsecref}[1]{\ref{sec:#1}~節 (\pageref{sec:#1}~ページ)}
\newcommand*{\fullfigref}[1]{\ref{fig:#1} (\pageref{fig:#1}~ページ)}
\newcommand*{\fulltabref}[1]{表~\ref{tab:#1} (\pageref{tab:#1}~ページ)}
\newcommand*{\fulequiref}[1]{式~\ref{equ:#1} (\pageref{equ:#1}~ページ)}

```

以上のようなマクロを作成しておけば入力が先程よりも簡単になるでしょう。

```

\section{減法混色}\seclab{submixcolor}
\fullsecref{addmixcolor}ではほげ。

```

L^AT_EX で相互参照を使う機会は 1 回以上あると思いますので (この冊子の例を自分で入力するなど), この節で紹介したものをマクロパッケージ `myref.sty` としてまとめておくとも便利かもしれません^{*1}.

▼ 6.6.1 相互参照に関わる L^AT_EX の警告

コマンドプロンプトやシェルで表示される **LaTeX Warning:** の後に以下に示すような警告が表示されていると, 相互参照に関する問題が解消されていないことを示します.

Label 'key' multiply defined というのは `\label` 命令で同じラベル名を持つラベルを定義しているということです. ラベルの重複がありますので, 該当するラベルに別の名前を付けます.

Reference 'key' on page n undefined という警告が表示されたのならばラベル名が定義されていないことになります.

Label(s) may have changed. Return to get cross-references right. が表示されたらラベルの値が変更されたということなので, もう 1 度タイプセットをします. この作業は 1 度で終わらないこともあるのでメッセージが表示されなくなるまでタイプセットを繰り返すこともあります.

ラベルに関する問題はラベルの参照する名前などのスペルミスなども考えられます.

^{*1} <http://tex.dante.jp/> に置くことにします.

第 7 章

数式の書き方

L^AT_EX は T_EX をベースにした組版システムなので数式の組版が得意です。この章では基本的な数式の出力の仕方をご紹介します。数式は通常の文章とは異なった組版が行なわれます。そのため、思わぬ部分でミスをしてしまう可能性がありますので、この章は注意深く読んでください。

7.1 はじめに

L^AT_EX における数式の組み立てではグルーピングが重要です。修飾される要素を明確に区別します。数式は普通の文章とは違い数式環境に記述します。数式は文章とは異なり、変数、数学記号、演算子、分数などの特殊な記述をしなければならないために、明示的に「ここが数式である」と宣言する必要があります。文章の部分をテキストモード、数式を含む部分を数式モードと呼びます。数式モードはどこから数式をはじめてどこまで数式にするかという始点と終点を決める必要もあります。数式モードでは以下の制約があります。

- 空白や改行は常に一つのスペースとして扱われます。通常は L^AT_EX 側が自動で空白を挿入しますがユーザーが明示的に空白を挿入することもできます。
- 空行は作成しません。一つの式に対して一つの段落を書くことができます。
- 半角英字はすべて指示がない限り数式イタリック体 (*math italic*) になり、自動的に空白が調節されます。

7.2 数式の出力

数式は段落の中に挿入する文中数式と別行に挿入する別行数式の 2 種類があります。別行数式には番号付きで別行に挿入する `equation` 環境と複数行の番号付き数式を出力する `eqnarray` 環境などがあります。

▼ 7.2.1 文中数式

文中数式の出力には 3 通りあります。

```
$数式$
\<数式\>
\begin{math}数式\end{math}
```

どれも同じような動作をしますが、‘\<数式\>’で囲むものが簡単ですのでこれだけ使えば良いでしょう。

math 環境などは記述量が増えるので使わなくても構いませんが、あまりに数式が長くなり見づらいときには math 環境で入れ子にするとすっきりするかも知れません。

\$a\$ の 2 乗と \$b\$ の 2 乗を足したものは \$c\$ の 2 乗

に等しいということは $a^2 + b^2 = c^2$ \)

と表せるが \LaTeX では

$\begin{math} a^2 + b^2 = c^2 \end{math}$

と書くこともできる。

a の 2 乗と b の 2 乗を足したものは c の 2 乗に等

しいということは $a^2 + b^2 = c^2$ と表せるが \LaTeX

では $a^2 + b^2 = c^2$ と書くこともできる。

上記の例においてハット ‘^’ は添え字の上付きの機能を持っています。

▼ 7.2.2 グループینگ

変数 a の $x + y$ 乗を出力するために \LaTeX では一塊の要素を波括弧でグループिंगします。ここではべき乗を例にとって見てみましょう。

$\backslash(a^x+y \backslashneq a^{\{x+y\}} \backslash)$

$a^x + y \neq a^{x+y}$

グループイングによって数式の要素を一つのグループにします。数式環境に限りませんが \LaTeX では一つにしたい要素をグループとして扱い、波括弧でグループ化を行います。

▼ 7.2.3 別行数式

数式を別行に立てる方法は \LaTeX では主に 3 通りあります。

```
$$数式$$
\[数式\]
\begin{displaymath} 数式\end{displaymath}
```

これら三つの命令の前後で自動的に改行が入り新しい行から数式が出力されます。両方とも数式を中央揃えで表示します。数式を左揃えにしたければ文書クラスファイルのオプションに *fleqn* を指定します。上記の文中数式と同じで $\backslash[数式\backslash]$ だけを使ったほうが簡単です。displaymath 環境は記述量が増えるので使わなくても構いません。あまりに数式が長くなったときなどには使えるでしょう。

別行立て数式は $\backslash[$

$c^2 = a^2 + b^2$

$\backslash]$ のように自動的に中央揃えになります。

別行立て数式は

$c^2 = a^2 + b^2$

のように自動的に中央揃えになります。

別行立て数式は

```
\begin{displaymath}
a^2 + b^2 = c^2
\end{displaymath}
```

と書くこともできます。

別行立て数式は

$$a^2 + b^2 = c^2$$

と書くこともできます。

▼ 7.2.4 番号付き数式

文書の中で参照するだろうと思われる数式には番号を付けます。そのような数式を番号付き数式と呼び、数式が 1 行の場合は `equation` 環境で出力することができます。

```
\begin{equation}
数式\label{\ラベル}
\end{equation}
```

`equation` で囲むことにより 1 行の番号付きの数式を出力することが出来ます。番号付きの数式は基本的にラベルを貼ることが出来ます。ラベルの参照の仕方は 6.5 節を参照してください。

```
\begin{equation}
a^2 + b^2 = c^2 \label{eq:equ}
\end{equation}
```

$$a^2 + b^2 = c^2 \quad (7.1)$$

式`(\ref{eq:equ})`より`c^2`は`a^2+b^2`に等しい。

式 (7.1) より c^2 は $a^2 + b^2$ に等しい。

▼ 7.2.5 複数行数式

```
\begin{eqnarray*}
左辺 \& (=) \& 右辺\\
左辺 \& (=) \& 右辺
\end{eqnarray*}
```

流れのある複数行の数式や証明などでイコール '=' の位置を揃えるときは `eqnarray*` 環境を使用し、これを複数行数式と呼びます。この環境は任意の行数で 3 列の行列に似ています。必ず 1 行にはアンド '&' が二つ、行の終わりには改行 '\\ ' を書きます。ただし最終行には改行を入れません。また各列における成分は省略することが可能です。

```
\begin{eqnarray*}
f(x) \& = \& x^2 \quad \\
f'(x) \& = \& 2x
\end{eqnarray*}
```

$$\begin{aligned} f(x) &= x^2 \\ f'(x) &= 2x \end{aligned}$$

▼ 7.2.6 複数行番号付き数式

後から参照するだろう複数行の数式には番号付けを行います。これを複数行番号付き数式と呼び、`eqnarray` 環境を使って記述します。書式は `eqnarray*` と同じです。ラベ

ルは1行ごとに改行‘\\’の前に貼ることが出来ます．また番号を出力したくない行は \nonumber 命令によって番号を振らないこともできます．

```
\begin{eqnarray}
```

```
f(x)      &=& x^2 \label{eq1}\\
```

$$f(x) = x^2 \quad (7.2)$$

```
f'(x)     &=& 2x \label{eq2}\\
```

$$f'(x) = 2x \quad (7.3)$$

```
\int f(x)dx&=& x^3/3+C\nonumber
```

$$\int f(x)dx = x^3/3 + C$$

```
\end{eqnarray}
```

式~(\ref{eq1}) を微分したものが

式 (7.2) を微分したものが式 (7.3) である．

式~(\ref{eq2}) である．

複数行数式はすでに数式モードになっていますのでそれをさらに数式環境で囲むなどの記述はしません．最終行に改行を入れないでください．

7.3 書体の変更

数式では書体の変更が必要になると思います．例えば行列を表すものはボード体に変更し数式中で文字を表示するときがあるでしょう．そのようなときは書体変更用のコマンドを使います．数式中では通常のテキストモードで使う書体変更コマンドは使えませんので、数式の書体変更用のコマンドを使います．

数式中でしか使用できない書体用コマンドは表 7.1 の通りです．

表 7.1 数式モードにおける書体の変更

書体	命令	出力
標準の書体	\mathnormal	<i>ABCabc</i>
ローマン体	\mathrm	ABCabc
サンセリフ体	\mathsf	ABCabc
タイプライタ体	\mathtt	ABCabc
ボード体	\mathbf	ABCabc
イタリック体	\mathit	<i>ABCabc</i>
カリグラフィック体	\mathcal	<i>ABC</i>

```
\begin{displaymath}
```

```
\int f(x)dx \neq \int f(x)\mathrm{d}x
```

$$\int f(x)dx \neq \int f(x)dx$$

```
\end{displaymath}
```

行列を表現するのにブラックボードボード体（黒板太字書体）を使うことがあるそうです．これは文字が白抜きになりボード体よりも行列であることが分かりやすくなっています．これを使うには amssymb を読み込みます．

数式中で通常のテキストを使いたいときは amsmath パッケージを読み込み \text 命令を使います．命令は表 7.2 となります．

表 7.2 amssymb による数式書体の拡張

書体	命令	出力
フラクトゥール体	<code>\mathfrak</code>	$\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{a}\mathfrak{b}\mathfrak{c}$
ブラックボードボード体	<code>\mathbb</code>	$\mathbb{A}\mathbb{B}\mathbb{C}$
数式内テキスト	<code>\text</code>	テキスト

$$x \in \mathbf{R} \neq x \in \mathbb{R}$$

$$f(x)=1/(1+g(x)), (x=3\text{とする})$$

$$x \in \mathbf{R} \neq x \in \mathbb{R}$$

$$f(x) = 1/(1 + g(x)), (x = 3 \text{ とする})$$

7.4 数式における空白の調節

数式モードでは入力した半角空白が反映されません。L^AT_EX は数式モードでは自動的に隣り合う数式要素（アトム）から挿入すべき空白を決めています。ですがユーザが空白を調節したほうが正しい表記になるときがあります。ユーザー側で空白を調節するため表 7.3 のコマンドを使います。積分 ' \int ' や全微分 ' dx ' のあいだにはユーザーが空白を入れると意味的に正しくなります。

表 7.3 数式における空白の制御

空白の大きさ	命令	入力例	出力例
空白なし	<code>\,</code>	<code>dx\,dy</code>	$dx\,dy$
かなり小さい空白	<code>\,</code>	<code>dx\, dy</code>	$dx\, dy$
小さい空白	<code>\:</code>	<code>dx\: dy</code>	$dx\, dy$
少し小さい空白	<code>\;</code>	<code>dx\; dy</code>	$dx\, dy$
半角の空白	<code>_</code>	<code>dx_dy</code>	$dx\, dy$
全角の空白	<code>\quad</code>	<code>dx\quad dy</code>	$dx\, dy$
全角の 2 倍の空白	<code>\qquad</code>	<code>dx\qquad dy</code>	$dx\, dy$
負の小さい空白	<code>\!</code>	<code>dx\!dy</code>	$dx\,dy$

$$\int \int f(x) dx dy \neq \int \int f(x) \, dx \, dy$$

$$\int \int f(x) dx dy \neq \int \int f(x) \, dx \, dy$$

7.5 基本的な数式コマンド

数式を書く環境を理解したら実際にそこに記述する記号などを覚えることになります。

▼ 7.5.1 添え字

L^AT_EX での添え字の入力は簡単です。

値^{上付き}値_{下付き}

添え字には上付きと下付きの2種類があります。これらの添え字を使うにはグルーピングの必要があります。1文字だけの添え字のときに丸括弧は必要ありませんが、添え字にしたいものが複数のときはグルーピングの処理が必要です。表7.4で例を示しますので参考にしてください。添え字をつけるときに上付きと下付きの順番は関係ありません。添え

表 7.4 添え字の使い方の例

意味	命令	出力	意味	命令	出力
右上	<code>x^{a+b}</code>	x^{a+b}	左上	<code>{ }^{a+b}x</code>	^{a+b}x
右下	<code>x_{a+b}</code>	x_{a+b}	左下	<code>{ }_{a+b}x</code>	$_{a+b}x$
右上と右下	<code>x^{a+b}_{c+d}</code>	x^{a+b}_{c+d}	左上と左下	<code>{ }^{a}_{b}x</code>	$^{a}_{b}x$
右上の右上	<code>x^{a^{b}}</code>	x^{a^b}	左下と右下	<code>{ }_{a}x_{b}</code>	$_{a}x_{b}$

字は何もないものに対しても添えることが可能です。表7.4でもその方法がとられています。

`\({ }^{a+b}_{x+y}A^{a+b}_{x+y} \)` $^{a+b}_{x+y}A^{a+b}_{x+y}$

ハット ‘`^`’ やアンダーバー ‘`_`’ は別の命令としても用意されています。上付きの`\sp` と下付きの`\sb` 命令を使うと良いでしょう。

`\(A^4_3 \neq A^{\sp4}_{\sb3} \)` $A^4_3 \neq A^{\sp4}_{\sb3}$

以上のような方法では左側に添え字を付けるときにうまくいかない場合がありますので、Harald harders 氏による `leftidx` パッケージを使います。

```
\leftidx{<左側添え字>}{<数式>}{<右側添え字>}
\ltrans{<数式>}
```

置換行列の上付き添え字は若干空白を抑えるために `\ltrans` 命令を使います。

```
\begin{eqnarray*}
{}_a{}^b\left(\frac{x}{y}\right)_c{}^d & & \\
\neq & \leftidx{{}_a{}^b}{\left(\frac{x}{y}\right)_c{}^d} & \\
{}^{\mathrm{t}}A & \neq & \ltrans{A}
\end{eqnarray*}
```

$${}_a\left(\frac{x}{y}\right)_c{}^d \neq {}_a\left(\frac{x}{y}\right)_c{}^d$$

$${}^{\mathrm{t}}A \neq {}^{\mathrm{t}}A$$

▼ 7.5.2 数学関数

数式モードでは自動的に英字がイタリック体になります。これは変数を表すためです。‘*d*’ と ‘d’ では数式では違う意味を持ちます。数学関数や極限などはローマン体、まっすぐ

な書体で書くのが慣わしです。L^AT_EX ではあらかじめそのような関数が定義されており、すぐに使える命令は表 7.5 の通りです。

表 7.5 主な数学関数

<code>arccos</code>	<code>\arccos</code>	<code>cot</code>	<code>\cot</code>	<code>exp</code>	<code>\exp</code>	<code>\liminf</code>	<code>\liminf</code>	<code>sec</code>	<code>\sec</code>
<code>arcsin</code>	<code>\arcsin</code>	<code>coth</code>	<code>\coth</code>	<code>gcd</code>	<code>\gcd</code>	<code>\limsup</code>	<code>\limsup</code>	<code>sin</code>	<code>\sin</code>
<code>arctan</code>	<code>\arctan</code>	<code>csc</code>	<code>\csc</code>	<code>hom</code>	<code>\hom</code>	<code>log</code>	<code>\log</code>	<code>sinh</code>	<code>\sinh</code>
<code>arg</code>	<code>\arg</code>	<code>deg</code>	<code>\deg</code>	<code>inf</code>	<code>\inf</code>	<code>max</code>	<code>\max</code>	<code>sup</code>	<code>\sup</code>
<code>cos</code>	<code>\cos</code>	<code>det</code>	<code>\det</code>	<code>ker</code>	<code>\ker</code>	<code>min</code>	<code>\min</code>	<code>tan</code>	<code>\tan</code>
<code>cosh</code>	<code>\cosh</code>	<code>dim</code>	<code>\dim</code>	<code>lim</code>	<code>\lim</code>	<code>Pr</code>	<code>\Pr</code>	<code>tanh</code>	<code>\tanh</code>

`\[\cos^2x+\sin^2x \neq \cos^2x+\sin^2x \]`

$$\cos^2 x + \sin^2 x \neq \cos^2 x + \sin^2 x$$

また `\bmod` のように法を表すための命令もあります。

`\bmod{<文字列>}` (2 項演算子として)
`\pmod{<文字列>}`

`\(\mathrm{M} \bmod \mathrm{N} \neq \)`
`\mathrm{M} \pmod \mathrm{N} \)`

$$M \bmod N \neq M \pmod N$$

▼ 7.5.3 大きさ可変の数学記号

数式中では修飾するものによって大きさの変わる記号があります。積分記号などがそれにあたります。主な大きさが可変な記号は表 7.6 の通りです。

表 7.6 大きさ可変の数学記号

種類	命令	出力例
分数	<code>\frac{<分子>}{<分母>}</code>	$\frac{\text{分子}}{\text{分母}}$
根号	<code>\sqrt{<値>}</code>	$\sqrt{\text{値}}$
添え字付き根号	<code>\sqrt[<根>]{<値>}</code>	$\sqrt[\text{根}]{\text{値}}$
添え字付き積分	<code>\int^{<上付き>}_{<下付き>}</code>	$\int_{\text{下付き}}^{\text{上付き}}$
添え字付き総和	<code>\sum^{<上付き>}_{<下付き>}</code>	$\sum_{\text{下付き}}^{\text{上付き}}$

`\begin{displaymath}`
`\int^b_a f(x)dx \neq`
`\sqrt{\frac{1}{f(x)}}`
`\end{displaymath}`

$$\int_a^b f(x)dx \neq \sqrt{\frac{1}{f(x)}}$$

```
\begin{displaymath}
\sqrt{\frac{1}{g(x)} + \sqrt{\int f(x)dx}}
\end{displaymath}
```

$$\sqrt{\frac{1}{g(x)} + \sqrt{\int f(x)dx}}$$

```
\begin{displaymath}
\frac{1}{g(x)} + \frac{1}{5x^2+8x+5}
\end{displaymath}
```

$$\frac{1}{g(x)} + \frac{1}{5x^2+8x+5}$$

`\sum` や `\int` などの添え字は上下に付く場合と右上と右下に付く場合があります。これを変更するには `\limits` と `\nolimits` を使います。

```
\limits
\nolimits
```

`\limits` を添え字を行うコマンドの前に置くと添え字される記号の上下に添え字を表示します。`\nolimits` はその反対のことをします。

```
\begin{eqnarray*}
\sum\nolimits_{k=0}^n k \neq \sum_{k=0}^n k \\
\int_a^b dx \neq \int\limits_a^b dx
\end{eqnarray*}
```

$$\sum_{k=0}^n k \neq \sum_{k=0}^n k$$

$$\int_a^b dx \neq \int_a^b dx$$

```
\begin{eqnarray*}
\lim\nolimits_{n \rightarrow 0} n \neq \lim_{n \rightarrow 0} n \\
\prod_{i=1}^n n \neq \prod\limits_{i=1}^n n
\end{eqnarray*}
```

$$\lim_{n \rightarrow 0} n \neq \lim_{n \rightarrow 0} n$$

$$\prod_{i=1}^n n \neq \prod_{i=1}^n n$$

▼ 7.5.4 区切り記号と括弧

\LaTeX における区切り記号 (括弧を含む) は何も指定しなければ大きさが変わりません。区切り記号は

- `\left` と `\right` 命令を使って大きさを変える。
- 区切り記号の大きさを指定する。

という二つの方法によって大きさを変更することもできます。

```
\begin{displaymath}
\left[ \Big(x+y\Big) \right]
\end{displaymath}
```

$$\left[\left(x+y \right) \right]$$

括弧で括られたり、区切られる要素に応じて大きさが変更できる区切り記号は表 7.7 となります。括弧などは要素を区切るための記号で、要素をきちんと括るべきです。 \LaTeX においては大きさが可変な区切り記号を用いてそれらを書き表します。`'\left'` 命令と

表 7.7 主な区切り記号

((\rfloor	↕	\updownarrow	{	\lbrace
))		\lfloor	↑	\Uparrow	}	\rceil
[[\arrowvert	↓	\Downarrow	[\lceil
]]		\Arrowvert	↕	\Updownarrow]	\rceil
{	\{		\Vert	\	\backslash	{	\lbrace
}	\}		\vert	>	\rangle	}	\rbrace
		↑	\uparrow	<	\langle		\lceil
	\	↓	\downarrow	}	\rangle		\rceil

* 大型の区切り記号です。

‘\right’ 命令を対で使うと括られた要素が適切な大きさの括弧で区切られます。 \left と \right には表 7.7 から記号を選ぶことによって、左右の区切りの対を自由に組み合わせられます。可変の括弧は修飾する式によって自動的に大きさを変更されるのでとても便利です。

```
\begin{displaymath}
\left( \frac{1}{1+\frac{1}{1+x}} \right)
\end{displaymath}
```

$$\left(\frac{1}{1 + \frac{1}{1+x}} \right)$$

```
\[ \left\{ \left\{ \frac{1}{x} + 1 \right\} + \left\{ \frac{1}{x^2} + 2 \right\} \right\}
\right\}
```

$$\left\{ \left\{ \left(\frac{1}{x} + 1 \right) + \left(\frac{1}{x^2} + 2 \right) \right\} \right\}$$

```
\begin{displaymath}
\left\uparrow \int f(x) dx \right.
+\left\lgroup \int g(x) dx \right\rgroup
\end{displaymath}
```

$$\uparrow \int f(x) dx + \left[\int g(x) dx \right]$$

自分で括弧の大きさを指定することもできます。大きさを指定した場合はそれ以上括弧の大きさが変わりませんので注意が必要です (表 7.8)。

```
\begin{displaymath}
\Biggl| \int f(x) dx \Biggr|
\Biggm/ \int g(x) dx \Biggr|
\end{displaymath}
```

$$\left\| \left(\int f(x) dx \right) / \left(\int g(x) dx \right) \right\|$$

表 7.8 を見ると分かると思いますが、括弧、いわゆる区切り記号に対して \big や \Big を付けるとその区切り記号を特定の倍率で拡大するという機能があります。左側を区切るには \bigl 類を、関係子としての区切り記号は \bigr 類を、右側を区切る記号には

表 7.8 括弧の大きさを指定する例

/	/	(())			\	\
/	\big/	(\bigl()	\bigr)		\bigm		\bigm\
/	\Big/	(\Bigl()	\Bigr)		\Bigm		\Bigm\
/	\bigg/	(\biggl()	\biggr)		\biggm		\biggm\
/	\Bigg/	(\Biggl()	\Biggr)		\Biggm		\Biggm\

`\bigr` 類を, 特に指定しないならば `\big` 類を使うようにします. 上記の `\big` 類を使った例と `\left` と `\right` による例を見比べてください.

$$\frac{\int f(x) \, dx}{\int g(x) \, dx}$$

片方だけに区切り記号があれば良いときはピリオド ‘.’ でいずれかの記号を省略できます。

$$\left(\int f(x)dx + \int g(x)dx\right)$$

▼ 7.5.5 行列

LaTeX における行列は array 環境中に記述します。array 環境はそのままでは数式にはならず math 環境や $\backslash[]$ の中に入れたり $\$$ の中に入れてあげます。array 環境の基本的な使い方は

```
\begin{array}{c} \text{列指定子} \\ a_{11} \quad \& \quad \dots \quad \& \quad a_{1n} \quad \backslash \\ \vdots \quad \& \quad \ddots \quad \& \quad \vdots \quad \backslash \\ a_{m1} \quad \& \quad \dots \quad \& \quad a_{mn} \\ \end{array}
```

ということに m 行 n 列の行列を書きます．ここでアンド '&' は成分 (要素) の区切りを意味し，'\' は行の終わりを意味しています．括弧は必要ならば前述の区切り記号で括弧することもできます．表と行列は基本的に同じ構造で，縦の罫線も横の罫線も入れることができます．

`\begin{array}{列数と縦罫線の指定}`

の部分では 4 列あるならば

```
\begin{array}{lc|cr}
```

のようにします．このときの‘l’，‘c’，‘r’は行列の中の要素の配置場所を指定するものです．真ん中にはテキストバー‘|’があります，これは縦方向の罫線を表しています．このような記号を列指定子と呼びます．array環境中で指定できる列指定子は表 7.9 となります．array環境は入れ子にすることも出来ます．行列の中に行列を書いたりすることも出来ます．

表 7.9 array 環境の主な列指定子

列指定子	意味
l	行列の縦 1 列を左揃えにする
c	行列の縦 1 列を中央揃えにする
r	行列の縦 1 列を右揃えにする
	縦の罫線を引く
	縦の 2 重罫線を引く
@{表現}	表現を縦 1 列追加します
p{長さ}	ある列の幅の長さを直接指定します
*{回数}{項目}	回数分だけ項目を繰り返す．

```
\[ \left( \begin{array}{cc}
a & b \\ c & d \end{array} \right) \]
```

$$\left(\begin{array}{cc} a & b \\ c & d \end{array} \right)$$

横方向に行列が続く場合があるため array 環境の最後の行に改行は入れません．

```
\[ \left( \begin{array}{*{2}{c}}
a & b \\ c & d \end{array} \right)
\left( \begin{array}{c} m \\ n \end{array} \right) =
\left( \begin{array}{c} am+bn \\ cm+dn \end{array} \right) \]
```

$$\left(\begin{array}{cc} a & b \\ c & d \end{array} \right) \left(\begin{array}{c} m \\ n \end{array} \right) = \left(\begin{array}{c} am+bn \\ cm+dn \end{array} \right)$$

array 環境には次に示すような場合分けを行う使い方もあります．

```
\[ f(x)= \left\{ \begin{array}{cl}
x & (x > 0) \\
0 & (x = 0) \\
-x & (x < 0) \end{array} \right. \]
```

$$f(x) = \left\{ \begin{array}{cl} x & (x > 0) \\ 0 & (x = 0) \\ -x & (x < 0) \end{array} \right.$$

水平に罫線などを入れたりするときには `\hline` , 要素の中で縦の罫線を引くときには `\vline` などを使います (表 7.10) . 罫線などの使い方は以下の例を見てください .

表 7.10 array 環境中での罫線の命令

命令	意味
<code>\hline</code>	横に引けるだけの罫線を引きます
<code>\hline\hline</code>	引けるだけの 2 重の横罫線を引きます
<code>\vline</code>	要素の中で引けるだけの縦罫線を引きます
<code>\cline{<範囲>}</code>	要素の罫線を行の範囲を指定して引きます
<code>\multicolumn{<数値>}{<列指定子>}{<要素>}</code>	行を繋げて列指定子通りに出力します

```
\begin{displaymath}
\begin{array}{llc} \hline
\multicolumn{3}{c}{f(x)} \\
g(x) & h(x) & i(x) \\
j(x)+k(x)+l(x)+m(x)+ & o(x) & p(x) \\
\end{array}
\end{displaymath}
```

$$\begin{array}{ccc} f(x) \\ g(x) & h(x) & i(x) \\ j(x) + k(x) + l(x) + m(x) + & o(x) & p(x) \end{array}$$

array 環境の簡易版として行列作成用の `\matrix` と丸括弧を付ける `\pmatrix` と `\matrix` にラベルも付けられる `\bordermatrix` などの命令があります . ただし `\matrix` 命令と `\pmatrix` に関しては amsmath パッケージの matrix 環境や `\pmatrix` 環境を使った方が良いでしょう .

```
\[ \begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} a & b & c \end{pmatrix}
```

$$\begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} a & b & c \end{pmatrix}$$

`\bordermatrix` 環境の括弧では各成分を区切るにはアンド ‘&’ を使い , 行の終わりにには ‘\cr’ 命令を使います .

```
\[ A=\bordermatrix{
& 1 & 2 \cr
1 & a & b \cr
2 & c & d } \]
```

$$A = \begin{pmatrix} 1 & 2 \\ a & b \\ c & d \end{pmatrix}$$

7.6 表示形式の調整

数式を記述する各環境において自動的に各要素の大きさが決められます . 文中数式での分数は $\frac{a}{b}$ という出力になりますが , これでは少し小さいので $\frac{a}{b}$ としたいときがあると思います . そのようなときはユーザーが表示形式を変更するには表 7.11 の命令が使えます . あまり多用すると段落のあいだが空きすぎて逆に見栄えが悪くなるのである程度長い数式を文中に入れているときは別行立てにするのが良い方法です . また文中の数式に限りませんが , 分数は $\frac{a}{b}$ と書くよりも a/b とするほうが一般的で見やすいのでスラッシュによる表記にしたほうが良いでしょう .

表 7.11 数式の表示形式の変更

命令	出力形式	例 ($\frac{a}{b}$)
<code>\displaystyle</code>	別行立て形式	$\frac{a}{b}$
<code>\textstyle</code>	文中数式形式	$\frac{a}{b}$
<code>\scriptstyle</code>	添え字形式	$\frac{a}{b}$
<code>\scriptscriptstyle</code>	添え字の中の添え字形式	$\frac{a}{b}$

`\(f(x)\)` の不定積分 `\(\int f(x)dx\)` と `\(\displaystyle \int f(x)dx\)` は \LaTeX では少し違うし分数は $\frac{a}{b}$ と書くよりも a/b と書くほうが一般的である .

$f(x)$ の不定積分 $\int f(x)dx$ と $\int f(x)dx$ は \LaTeX では少し違うし分数は $\frac{a}{b}$ と書くよりも a/b と書くほうが一般的である .

```
\[ \frac{1}{1+\frac{1}{1+\frac{1}{1+x}}}
\neq \frac{1}{\displaystyle 1+
\frac{1}{\displaystyle 1+
\frac{1}{1+x}}} \]
```

$$\frac{1}{1+\frac{1}{1+\frac{1}{1+x}}} \neq \frac{1}{1+\frac{1}{1+\frac{1}{1+x}}}$$

```
\( \int_a^b f(x)dx \neq
\displaystyle \int_a^b g(x)dx
\)
```

$$\int_a^b f(x)dx \neq \int_a^b g(x)dx$$

7.7 数式モード中の記号

記号の中には数式モード中でしか使えないものがほとんどです . 以下の記号は`\(\)`で囲むなど , 数式環境の中で使用しないと **! Missing \$ inserted.** のようなエラーが表示されます .

▼ 7.7.1 ギリシャ文字

数式中の変数ならびに定数にはギリシャ文字を使うのが一般的です . ギリシャ小文字は表 7.12 , 小文字の変体文字は表 7.13 , 大文字は表 7.14 となります . ギリシ

表 7.12 ギリシャ小文字

α	<code>\alpha</code>	η	<code>\eta</code>	ν	<code>\nu</code>	τ	<code>\tau</code>
β	<code>\beta</code>	θ	<code>\theta</code>	ξ	<code>\xi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	o	<code>o</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	π	<code>\pi</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ρ	<code>\rho</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>

ヤ小文字においてオミクロン ‘ \omicron ’ だけはアルファベットの ‘ o ’ と同じため特別に記号が用意されていません。逆に ‘ $\backslash o$ ’ は文中で使うべき記号であり、この命令を数式中で使うと **LaTeX Warning: Command $\backslash o$ invalid in math mode on input line 30.** のように警告が表示されます。

```
\begin{eqnarray*}
\cos^2\theta+\sin^2\theta & \neq & \\
\cos^2x + \sin^2x & & \cos^2\theta + \sin^2\theta \neq \cos^2x + \sin^2x
\end{eqnarray*}
```

表 7.13 ギリシャ小文字の変体文字

ε	<code>\varepsilon</code>	ϑ	<code>\vartheta</code>	ϖ	<code>\varpi</code>
ϱ	<code>\varrho</code>	ς	<code>\varsigma</code>	φ	<code>\varphi</code>

表 7.14 ギリシャ大文字

A	<code>\mathrm{A}</code>	H	<code>\mathrm{H}</code>	N	<code>\mathrm{N}</code>	T	<code>\mathrm{T}</code>
B	<code>\mathrm{B}</code>	Θ	<code>\Theta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>
Γ	<code>\Gamma</code>	I	<code>\mathrm{I}</code>	O	<code>\mathrm{O}</code>	Φ	<code>\Phi</code>
Δ	<code>\Delta</code>	K	<code>\mathrm{K}</code>	Π	<code>\Pi</code>	X	<code>\mathrm{X}</code>
E	<code>\mathrm{E}</code>	Λ	<code>\Lambda</code>	P	<code>\mathrm{P}</code>	Ψ	<code>\Psi</code>
Z	<code>\mathrm{Z}</code>	M	<code>\mathrm{M}</code>	Σ	<code>\Sigma</code>	Ω	<code>\Omega</code>

ギリシャ大文字でもアルファベットと同じ文字は特別な記号が用意されておりません。ギリシャ小文字と同じようにオミクロン ‘ $\backslash o$ ’ を数式中で使うと次のような警告が表示されます。

LaTeX Warning: Command $\backslash o$ invalid in math mode on input line 40.

さらにギリシャ大文字の $A, B, E, Z, H, I, K, M, N, O, P, T, X$ はそのままではイタリック体となって変数を意味してしまいますので定数としてのギリシャ大文字を出力するためには `\mathrm` を使います。

```
\begin{eqnarray*}
A & & \neq & & \mathrm{A} \\
F(x)+C & & \neq & & F(x)+\mathrm{C} \\
\mathrm{diff} & & \neq & & \mathrm{diff}
\end{eqnarray*}
```

$$A \neq A$$

$$F(x) + C \neq F(x) + C$$

$$diff \neq diff$$

▼ 7.7.2 関係子や演算子などの数学記号

表 7.15 関係子

以下のコマンドの前に \not コマンドを付ければその関係子の否定になります

\leq	<code>\le</code>	\in	<code>\in</code>	\supseteq	<code>\sqsupseteq</code>	\neq	<code>\neq</code>
\prec	<code>\prec</code>	\notin	<code>\notin</code>	\dashv	<code>\dashv</code>	\doteq	<code>\doteq</code>
\preceq	<code>\preceq</code>	\geq	<code>\ge</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>
\ll	<code>\ll</code>	\succ	<code>\succ</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
\subset	<code>\subset</code>	\succeq	<code>\succeq</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\subseteq	<code>\subseteq</code>	\gg	<code>\gg</code>	\simeq	<code>\simeq</code>	\mid	<code>\mid</code>
\sqsubseteq	<code>\sqsubseteq</code>	\supset	<code>\supset</code>	\asymp	<code>\asymp</code>	\parallel	<code>\parallel</code>
\vdash	<code>\vdash</code>	\supseteq	<code>\supseteq</code>	\approx	<code>\approx</code>	\bowtie	<code>\bowtie</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\cong	<code>\cong</code>		

表 7.16 2 項演算子

\pm	<code>\pm</code>	\cdot	<code>\cdot</code>	\setminus	<code>\setminus</code>	\ominus	<code>\ominus</code>
\mp	<code>\mp</code>	\cap	<code>\cap</code>	\wr	<code>\wr</code>	\otimes	<code>\otimes</code>
\times	<code>\times</code>	\cup	<code>\cup</code>	\diamond	<code>\diamond</code>	\oslash	<code>\oslash</code>
\div	<code>\div</code>	\uplus	<code>\uplus</code>	\bigtriangleup	<code>\bigtriangleup</code>	\odot	<code>\odot</code>
\ast	<code>\ast</code>	\sqcap	<code>\sqcap</code>	\bigtriangledown	<code>\bigtriangledown</code>	\bigcirc	<code>\bigcirc</code>
\star	<code>\star</code>	\sqcup	<code>\sqcup</code>	\triangleleft	<code>\triangleleft</code>	\dagger	<code>\dagger</code>
\circ	<code>\circ</code>	\vee	<code>\vee</code>	\triangleright	<code>\triangleright</code>	\ddagger	<code>\ddagger</code>
\bullet	<code>\bullet</code>	\wedge	<code>\wedge</code>	\oplus	<code>\oplus</code>	\amalg	<code>\amalg</code>

表 7.17 大型演算子

これらは大きさが可変です

\sum	<code>\sum</code>	\oint	<code>\oint</code>	\bigvee	<code>\bigvee</code>	\bigoplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\bigcup	<code>\bigcup</code>	\bigwedge	<code>\bigwedge</code>	\bigotimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\bigcap	<code>\bigcap</code>			\bigodot	<code>\bigodot</code>
\int	<code>\int</code>	\bigsqcup	<code>\bigsqcup</code>			\biguplus	<code>\biguplus</code>

`\(\vec{a} + \vec{b} \neq \vec{a + b} \)`
`\neq \overrightarrow{a + b} \)`

$\vec{a} + \vec{b} \neq \vec{a + b} \neq \overrightarrow{a + b}$

表 7.18 小さいアクセント

これらの小さいアクセントは大きさが変わりません

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\breve{a}	<code>\breve{a}</code>	\acute{a}	<code>\acute{a}</code>
\grave{a}	<code>\grave{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\bar{a}	<code>\bar{a}</code>	\dot{a}	<code>\dot{a}</code>
\ddot{a}	<code>\ddot{a}</code>	\vec{a}	<code>\vec{a}</code>				

表 7.19 大きいアクセント

大きいアクセントは大きさが可変です

$\overline{m+M}$	<code>\overline</code>	$\overbrace{m+M}$	<code>\overbrace</code>
$\underline{m+M}$	<code>\underline</code>	$\underbrace{m+M}$	<code>\underbrace</code>
$\overleftarrow{m+M}$	<code>\overleftarrow</code>	$\widehat{m+M}$	<code>\widehat</code>
$\overrightarrow{m+M}$	<code>\overrightarrow</code>	$\widetilde{m+M}$	<code>\widetilde</code>

```
\begin{displaymath}
\overbrace{a+b+c+d+e+f+g}^{\sim{h+i+j+k}+}
\underbrace{l+m+n}_{\sim{o+p+q}}
\end{displaymath}
```

$$\overbrace{a+b+c+d+e+f+g}^{h+i+j+k} + \underbrace{l+m+n}_{o+p+q}$$

表 7.20 矢印

\leftarrow	<code>\leftarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\hookleftarrow	<code>\hookleftarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\rightleftharpoons	<code>\rightleftharpoons</code>
\leftharpoonup	<code>\leftharpoonup</code>	\hookrightarrow	<code>\hookrightarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoonup	<code>\rightharpoonup</code>	\Updownarrow	<code>\Updownarrow</code>
\longleftarrow	<code>\longleftarrow</code>	\rightharpoondown	<code>\rightharpoondown</code>	\Uparrow	<code>\Uparrow</code>
\Longleftarrow	<code>\Longleftarrow</code>	\uparrow	<code>\uparrow</code>	\nearrow	<code>\nearrow</code>
\rightarrow	<code>\rightarrow</code>	\Uparrow	<code>\Uparrow</code>	\swarrow	<code>\swarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\downarrow	<code>\downarrow</code>	\searrow	<code>\searrow</code>
\mapsto	<code>\mapsto</code>	\Downarrow	<code>\Downarrow</code>	\nwarrow	<code>\nwarrow</code>

```
\begin{displaymath}
(p\rightarrow r)\vee
(q\rightarrow s)
\end{displaymath}
```

$$(p \rightarrow r) \vee (q \rightarrow s)$$

```
\[ \forall x \forall y (P(x,y) \vee (f(x) \wedge g(x))) \]
```

$$\forall x \forall y (P(x, y) \vee (f(x) \wedge g(x)))$$

表 7.21 特殊な数学記号

\aleph	<code>\aleph</code>	∂	<code>\partial</code>	\bot	<code>\bot</code>	\natural	<code>\natural</code>
\hbar	<code>\hbar</code>	∞	<code>\infty</code>	\angle	<code>\angle</code>	\sharp	<code>\sharp</code>
\imath	<code>\imath</code>	\prime	<code>\prime</code>	\triangle	<code>\triangle</code>	\clubsuit	<code>\clubsuit</code>
\jmath	<code>\jmath</code>	\emptyset	<code>\emptyset</code>	\forall	<code>\forall</code>	\diamondsuit	<code>\diamondsuit</code>
ℓ	<code>\ell</code>	∇	<code>\nabla</code>	\exists	<code>\exists</code>	\heartsuit	<code>\heartsuit</code>
\wp	<code>\wp</code>	\surd	<code>\surd</code>	\neg	<code>\neg</code>	\spadesuit	<code>\spadesuit</code>
\Re	<code>\Re</code>	$ $	<code> </code>	\backslash	<code>\backslash</code>		
\Im	<code>\Im</code>	\top	<code>\top</code>	\flat	<code>\flat</code>		

```
\( e^{j\theta}=\Re\{e^{j\theta}\}
+\Im\{e^{j\theta}\}
=\cos\theta+j\sin\theta\)
```

$$e^{j\theta} = \Re\{e^{j\theta}\} + \Im\{e^{j\theta}\} = \cos \theta + j \sin \theta$$

表 7.22 点

```
... \ldots | ... \cdots | \vdots \vdots | \ddots \ddots
```

```
\[ (a_0+a_1+\cdots+a_n)
\neq \{a_0,a_1,\ldots,a_n\} \]
```

$$(a_0 + a_1 + \cdots + a_n) \neq \{a_0, a_1, \dots, a_n\}$$

▼ 7.7.3 標準ではない数学記号

$\text{\LaTeX} 2_{\epsilon}$ からはこぼれた記号類を出力するためには, Frank Mittelbach 氏が作成した `latexsym` を読み込むと良いでしょう. すでに `amssymb` か `amsfonts` を読み込んでいるならば, そちらに定義されているので `latexsym` をさらに読み込まなくても良いです.

表 7.23 標準ではない数学記号

\mho	<code>\mho</code>	\Join	<code>\Join</code>	\Box	<code>\Box</code>	\Diamond	<code>\Diamond</code>
\leadsto	<code>\leadsto</code>	\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\lhd	<code>\lhd</code>
\unlhd	<code>\unlhd</code>	\rhd	<code>\rhd</code>	\unrhd	<code>\unrhd</code>		

7.8 定義や定理など

`\theorem` 命令を使うと新規に定義型や定理型の環境を作成できます.

```
\newtheorem{<名前>}{<ラベル>}[<親カウンタ>]
\newtheorem{<名前>}[<定義済みの環境>]{<ラベル>}
```

章や節などを通し番号の前に付けるにはその〈親カウンタ〉を表 6.2 から選びます．別々の環境で同じ通し番号を使いたい場合は〈定義済みの環境〉を指定します．具体的な使用例として

```
\newtheorem{Prob}{問題}[chapter]
\newtheorem{Exe}{例題}[Prob]
```

をプリアンプルに記述しておけば以下のように使えます．

```
\begin{Exe}\label{Hoge:ware}
```

この冊子は難しいか．答えは簡単だ．

```
\end{Exe}
```

▷ 例題 7.1 この冊子は難しいか．答えは簡単だ．

```
\begin{Prob}\label{Geho:yueni}
```

この冊子は有益かどうか考えよ．

▶ 問題 7.2 この冊子は有益かどうか考えよ．

```
\end{Prob}
```

例題 7.1 より問題 7.2 が導かれる．

例題~\ref{Hoge:ware}より

問題~\ref{Geho:yueni}が導かれる．

実際の出力は異なると思います．`\theorem` 命令は定理型や定義型の環境を作成するために作られたので日本語用には思うようにカスタマイズできないようです．

▼ 7.8.1 定理型環境のカスタマイズ

Frank Mittelbach 氏が作成した `theorem` は \LaTeX における `\theorem` 命令を拡張したパッケージです．このパッケージは例えば「定理型」や「定義型」だけでなく、「問題型」や「例題型」などの環境を作成するときに満足の行く出力になると思われます． \LaTeX に含まれる `amsthm` というパッケージもありますが Frank Mittelbach 氏が作成した `theorem` を使ったほうが便利だと思います．定理型の環境を新設するときは \LaTeX の `\theorem` 命令と同じように

```
\newtheorem{<環境名>}{<名前>}
```

によって行います．さらに章などの親カウンタに連動させたい場合は

```
\newtheorem{<環境名>}{<名前>}[<カウンタ名>]
```

のようにしますし，同系の環境を作成するときは

```
\newtheorem{<環境名>}[<同系の環境名>]{<名前>}
```

として定義します．`theorem` パッケージではさらにそれぞれの定理型環境の書式を以下の命令で変更できます．

```
\theoremstyle{<スタイル>}
```

```
\theorembodyfont{<書式>}
```

```
\theoremheaderfont{<書式>}
```

〈書式〉に対しては書体変更の宣言型の命令を使います．〈スタイル〉には以下の六つが使えます．

plain 標準の \theorem 命令と同じ書式にします。

break 〈名前〉を出力した後に改行をします。

margin 通し番号を余白に出力します。

change 通し番号と〈名前〉を入れ替えます。

marginbreak ‘margin’に付け加え、それを出力した後に改行します。

changebreak ‘change’に付け加え、それを出力した後に改行します。

theorem パッケージで「命題 2.1, 定義 2.2, 定理 2.3」のような環境を作成したければ次のようにします。

```
{\theorembodyfont{\normalfont}
\theoremheaderfont{\normalfont\gtfamily\bfseries}
\newtheorem{Exam}{命題}
\newtheorem{Refer}[Exam]{定義}
\newtheorem{Prob}[Exam]{定理}}
```

7.9 雑多なこと

不定積分を表現したり定積分を表現したりする場合を考えてみましょう。

```
%\usepackage{txfonts}
\[ \int f(x)dx + \int g(y)dy + \iint h(x,y)dx\,dy \]
```

$$\int f(x)dx + \int g(y)dy + \iint h(x,y)dx\,dy$$

この場合は新規に \intx や \iintxy を定義すると手間が省けるでしょう。

```
\newcommand*\intx[1]{\int#1dx}
\newcommand*\inty[1]{\int#1dy}
\newcommand*\iintxy[1]{\iint#1dx\,dy}
\[ \intx{f(x)} + \inty{g(y)} + \iintxy{h(x,y)} \]
```

$$\int f(x)dx + \int g(y)dy + \iint h(x,y)dx\,dy$$

ある線形微分方程式 $dy/dx + P(x)y = Q(x)$ の一般解を表現するために

```
\[ y = e^{-\int P(x)dx} \left\{ \int Q(x)e^{\int P(x)dx}dx + \mathrm{c} \right\}
```

$$y = e^{-\int P(x)dx} \left\{ \int Q(x)e^{\int P(x)dx}dx + c \right\}$$

というのを何回も書くのはエネルギーの無駄ですから、公式通りに新規に命令を作ると汎用的に $P(x)$ や $Q(x)$ を書くことができます。

```
\newcommand{\my}{%
\ensuremath{dy/dx+P(x)y=Q(x)}}
\newcommand{\mypq}[2]{\ensuremath{%
e^{\int\{#1\}dx}\left\{\int\{#2\}%
e^{\int\{#1\}dx}dx+\mathrm{c}\right\}}}
$P(x)=x^2+\pi$, $Q(x)=e^{-x}$とすると\my の
解$y$は\[\mypq{(x^2+\pi)}{e^{-x}}\] となる。
```

$$P(x) = x^2 + \pi, Q(x) = e^x \text{ とすると } dy/dx + P(x)y = Q(x) \text{ の解 } y \text{ は}$$

$$e^{\int (x^2+\pi)dx} \left\{ \int e^x e^{\int (x^2+\pi)dx} dx + c \right\}$$

となる。

何らかの数式が公式として確立している場合はそれをマクロとして作成しておくとう便利です。マクローリン展開やテイラー展開を毎回書くのは面倒ですから次のような使い方をすると良いでしょう。

```
\newcommand{\mac1}[2][x]{\ensuremath{%
  f(#2)+\frac{1}{1!}f'(#2)(#1-#2)+
  \frac{1}{2!}f''(#2)(#1-#2)^2+\cdots
  +\frac{1}{k!}f^{(k)}(#2)(#1-#2)^k
  +\cdots}}
\newcommand{\Mac1}[2][x]{\ensuremath{%
  \sum^{\infty}_{k=0}\frac{1}{k!}%
  f^{(k)}(#2)(#1-#2)^k}}
関数$f(z)$の$z=0$におけるテイラー展開は
\(\mac1[z]{0}\)であり\(\Mac1[z]{0}\)
となるので$z=0$における級数は[
  f(z)=\sum^{\infty}_{k=0}\frac{1}{k!}
  f^{(k)}(0)z^k
\] となり、これをマクローリン展開と呼ぶ。
```

関数 $f(z)$ の $z = 0$ におけるテイラー展開は $f(0) + \frac{1}{1!}f'(0)(z-0) + \frac{1}{2!}f''(0)(z-0)^2 + \cdots + \frac{1}{k!}f^{(k)}(0)(z-0)^k + \cdots$ であり $\sum_{k=0}^{\infty} \frac{1}{k!}f^{(k)}(0)(z-0)^k$ となるので $z = 0$ における級数は

$$f(z) = \sum_{k=0}^{\infty} \frac{1}{k!} f^{(k)}(0) z^k$$

となり、これをマクローリン展開と呼ぶ。

偏微分記号が多く出てくる数式を考えます。

```
\[ \frac{\partial\{f\}}{\partial\{x\}}
  +\frac{\partial^2\{f\}}{\partial\{x\}^2}
  +\frac{\partial^3\{f\}}{\partial\{x\}^3}
\]
```

$$\frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial x^2} + \frac{\partial^3 f}{\partial x^3}$$

毎回このように記述するのは疲れますので次のようにマクロを作成して用います。

```
\newcommand{\pdiff}[3][]{\ensuremath{%
  \frac{\partial^{#1}\{#2\}}
  {\partial\{#3\}^{#1}}}}
\[ \pdiff{f}{x}+\pdiff[2]{f}{x} \]
```

$$\frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial x^2}$$

このようにしても良いのですが、変数が二つ以上の場合は手動で対処します。

```
\newcommand{\pdiff}[3][]{\ensuremath{%
  \frac{\partial^{#1}\{#2\}}
  {\partial\{#3\}^{#1}}}}
\[ \pdiff[2]{f}{x} + \pdiff[2]{f}{xy} +
  \pdiff[2]{f}{y} \]
```

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial xy} + \frac{\partial^2 f}{\partial y^2}$$

`\partial` と `\frac` を乱雑に書くよりはこのほうがすっきりしているでしょう。

作成中の文書の分野を考えてあらかじめ公式の一部分をマクロとして作成するのも有効かも知れません。

▼ 7.9.1 記号の積み重ね

イコール ‘=’ のうゑに ‘def’ をのせて ‘ $\stackrel{\text{def}}{=}$ ’ のような記号を出したいときがあります。これには `\stackrel` という命令が使えます。二つ目の引数を二つ目の引数のうゑに載せて関係子を作ります。

```
\stackrel{<上の記号>}{<下の記号>}
```

```
\newcommand{\defeq}{\%
```

```
\stackrel{\mathrm{def}}{=}}
```

$$x \stackrel{\text{def}}{=} p(t) + q(t) + r(t)$$

```
\( x \defeq p(t)+q(t)+r(t) \)
```

記号の積み重ねとは少し違うのですが、次のような数式を出力するときもあるでしょう。この例では`\substack` という `amsmath` パッケージに含まれる命令を使っています。

```
\begin{displaymath}
```

```
\sum_{i=1}^l \sum_{j=1}^m \sum_{k=1}^n
```

```
p_i q_j r_k \neq \sum_{
```

```
\substack{i \leq 1 \leq l \\ j \leq 1 \leq m
```

```
\\ k \leq 1 \leq n}} p_i q_j r_k
```

```
\end{displaymath}
```

$$\sum_{i=1}^l \sum_{j=1}^m \sum_{k=1}^n p_i q_j r_k \neq \sum_{\substack{i \leq 1 \leq l \\ j \leq 1 \leq m \\ k \leq 1 \leq n}} p_i q_j r_k$$

7

▼ 7.9.2 記号の重ね合わせ

二つの記号を重ね合わせて新しい記号を作りたいときがあります。`\oalign` と `\crrc` 命令を組み合わせたうまくできます。

```
{\oalign{\cr\crrc{二つ目}}}
```

二つの記号の中で横幅の広いほうの幅が優先されます。二つの記号を中心に重ね合わせたいときは `\hss` という空白を挿入する命令を使います。さらに文字列に `\not` を使っても演算子の否定のようにはなりませんので

```
\newcommand{\cnot}[1]{\oalign{\crrc{\hss{#1}\hss}}}
```

のような定義をしておくと良いでしょう。スラッシュは全角を使っています。

```
\newcommand{\pile}[2]{\%
```

```
{\oalign{\crrc{#1}\crrc{#2}}}
```

```
\newcommand{\cpile}[2]{\oalign{\%
```

```
\hss{#1}\hss}\crrc{\hss{#2}\hss}}}
```

```
\newcommand{\cnot}[1]{\%
```

```
\oalign{\crrc{\hss{#1}\hss}}}
```

円記号 $\$ \pile Y = \$$ はこの $\$ \cpile Y = \$$ とは

別物で、`\cnot{A}` も `\pile/A` とは別物である。

円記号 \yen はこの \yen とは別物で、 \A も \A とは別物である。

▼ 7.9.3 数式の太字

何らかの理由である数式の一部や、ある数式全体を太字にすることがあります。方法はいくつか存在します。

- `\mathbf` 命令を使う。
- `\boldmath` と `\unboldmath` を使って太字かどうかを切り替える。

- amsmath に含まれる amsbsy パッケージの `\boldsymbol` 命令を使う。
- bm パッケージの `\bm` 命令を使う。

などがあります。これは使用している数式書体によっては使えないことがあります。txfonts や pxfonts を使うとなんら問題なく出力できます。一つ目の `\boldmath` と `\unboldmath` は数式モード中で使うことができません。

```
\(\mathbf{\int^a_b f(x)dx} \quad \backslash neq\)
\boldmath \(\int^a_b f(x)dx \quad \backslash neq\)
\unboldmath\(\int^a_b f(x)dx \backslash\)
```

$$\int_b^a f(x)dx \neq \int_b^a f(x)dx \neq \int_b^a f(x)dx$$

`\mathbf` の場合はギリシャ文字などの特定の記号しか太字にならないうえにイタリック体ではなくローマン体になってしまいます。もう少し局所的に使いたい場合は amsbsy の `\boldsymbol` を使います。

```
\(\mathbf{\int^a_b f(x)dx} \quad \backslash neq\)
\boldsymbol{\int^a_b f(x)dx} \quad \backslash neq\)
\int^a_b f(x)dx \backslash\)
```

$$\int_b^a f(x)dx \neq \int_b^a f(x)dx \neq \int_b^a f(x)dx$$

amsbsy を使うよりも bm パッケージの `\bm` を使うほうが安全です。

```
\(\mathbf{\int^a_b f(x)dx} \quad \backslash neq\)
\bm{\int^a_b f(x)dx} \quad \backslash neq\)
\int^a_b f(x)dx \backslash\)
```

$$\int_b^a f(x)dx \neq \int_b^a f(x)dx \neq \int_b^a f(x)dx$$

結論として `\bm` 命令を使うようにすると思い通りの結果になるのではないかと思います。

▼ 7.9.4 高さを揃える

ルート記号などを使っているとルートの高さが揃わずに見栄えが悪くなるときがあります。これには数式中でルートなどの高さを揃える `\mathstrut` 命令が使えます。

```
\[ \overline{\sqrt{a} + \sqrt{b}} \quad \backslash neq
\sqrt{\mathstrut a} + \sqrt{\mathstrut b} \quad \backslash neq
\sqrt{\mathstrut a} + \sqrt{\mathstrut b} \quad \backslash neq\]
```

$$\overline{\sqrt{a} + \sqrt{b}} \neq \sqrt{a} + \sqrt{b}$$

分かりづらいのですが実は高さのみならず、深さも `\mathstrut` によって自動的に調整されています。

もう少し高度な命令として `\phantom`、`\vphantom`、`\hphantom` の三つが用意されています。`\phantom` 命令は引数に与えられた要素だけの高さ、幅と深さを持った空白を作成します。`\vphantom` は引数に与えた要素の高さと同じ目には見えない箱を作成します。`\hphantom` はその横方向バージョンです。

```
\[ \sqrt{\int f(x)dx} + \sqrt{g(x)} \quad \backslash neq
\sqrt{\int f(x)dx} + \sqrt{\phantom{\int f(x)dx} g(x)} \quad \backslash neq
\vphantom{\int f(x)dx} \sqrt{g(x)} \quad \backslash neq\]
```

$$\sqrt{\int f(x)dx} + \sqrt{g(x)} \neq \sqrt{\int f(x)dx} + \sqrt{g(x)}$$

もう一つ `\smash` という命令もあり、これは引数に与えられた要素の高さと深さを 0 にする魔法のようなものです。 `\smash` と `\vphantom` を組み合わせると要素の幅はそのまま高さ・深さを 0 にしたうえで `\vphantom` で指定した高さ・深さの見えない箱を作成できるので、高さや深さを揃えるのに使えます。

```
\begin{displaymath}
\quad \underbrace{a+b}+\underbrace{i+j}\neq
\quad \underbrace{\smash{a+b}\vphantom{i+j}}
+\underbrace{i+j}
\end{displaymath}
```

$$\underbrace{a+b}+\underbrace{i+j}\neq\underbrace{a+b}+\underbrace{i+j}$$

▼ 7.9.5 スマートな分数の書き方

文中数式中で分数を出力する `\frac` 命令を使うと $\frac{a}{b}$ となります。このような分数の書き方はスマートではありません。 a/b と書くと一般的な文中の分数のスタイルとなります。

```
\begin{displaymath}
\quad \frac{\frac{a}{b}}{c}\neq\frac{a/b}{c}
\end{displaymath}
```

$$\frac{\frac{a}{b}}{c}\neq\frac{a/b}{c}$$

このような分数のスタイルは別行数式にも当てはまります。別行数式において分数を記述しており、その分母・分子上にさらに分数を書く、連分数を記述する場合などはスラッシュ ‘/’ による表記をするとスマートになります。ただしスラッシュによる表記では適宜丸括弧を補います。

```
\begin{displaymath}
\quad \frac{\frac{a-b}{c}}{d}\neq
\quad \frac{a-b/c}{d}\neq\frac{(a-b)/c}{d}
\end{displaymath}
```

$$\frac{\frac{a-b}{c}}{d}\neq\frac{a-b/c}{d}\neq\frac{(a-b)/c}{d}$$

```
\begin{displaymath}
\quad \frac{x+f(x)}{x-g(x)}\neq
\quad \frac{(x+f(x))/(x-g(x))}{x-g(x)}
\end{displaymath}
```

$$\frac{x+f(x)}{x-g(x)}\neq\frac{(x+f(x))/(x-g(x))}{x-g(x)}$$

▼ 7.9.6 数式モード中の空白と書体

数式用の環境では自動的に要素の前後の記号の種類になどにより空白が調節されますから意図していた結果と異なる場合があります。

```
\emph{diff}は\langle diff\rangle にはなりませんから
\l[ diff \neq \mathit{diff}. \r]
diff は diff にはなりませんから
diff \neq diff.
```

$$\textit{diff} \neq \textit{diff}.$$

‘diff’ という文字が全て数式中では変数と解釈され、それぞれ \LaTeX が適切だと思う空白を挿入してくれています。これから分かるように数式モード中ではユーザが明示的に空白を調節すると良い場合があります。

```
$10\times5,000=50,000$円になります！\par      10 × 5,000 = 50,000 円になります！
$10\times5{,}000=50{,}000$円になる？\par      10 × 5,000 = 50,000 円になる？
```

上記の例ではコンマ ‘,’ が恐らく何かの区切りとして解釈されたのでしょう、意図していたものよりも広がっています。同じように感嘆符 ‘!’ などは逆に空白が挿入されません。ですから `\,` 命令で若干の空きを挿入します。

```
\[ \frac{s! (q-1)! (r-2)!}{p! q! r!} \neq \frac{s! (q-1)! (r-2)!}{p! q! r!}
```

感嘆符 ‘!’ の例を見ると分かりますが数式モード中では斜体になっていません。このように数式モード中でも斜体にならない記号がいくつかあります。`\textit` では記号もイタリック体になりますが数式中の `\mathit` を使うといくつかの記号が斜体にならないばかりか、空白制御が行われません。

```
%\usepackage{amsmath}
\textit{This is text mode?!}\par      This is text mode?!
\(\mathit{Is\ this\ text\ mode?!})\par  Is this text mode?!
\(\mathit{Is this text mode?!})\par      Isthistextmode?!
\(\mathrm{Is this text mode?!})\par      Isthistextmode?!
\(\text{Is this text mode?!})\par        Is this text mode?!
\(\textit{Is this text mode?!})\par      Isthistextmode?!
```

いずれの場合も疑問符 ‘?’ はイタリック体にはなっていません。このように数式中では明示的にイタリック体に書体を変更する命令を使ってもローマン体のままの記号があります。

第 8 章

図表の貼り方

レポート・論文に図や表を取り入れることは読者の理解を助けることになります。この章では文書中にどのように図表を構成すればよいのかを解説します。

8.1 図表の基礎

▼ 8.1.1 一般的な取り決め

以下の説明は一般的なレポート・論文作成における図表の取り決めです。

図表の位置 一般的に、論文中において図表はページの上端か下端に出力します。関係文章よりも前出することがなければ、本文中に配置することも可能です。ただし、図表の前後に文章が 1 行だけ取り残されるようなことは避けるようにします。図表は中央揃えにします。このとき図表の左右に文章を流し込むこともありますが、原則として図表と本文を区別するために、左右に文章は記述してはいけません。

図表と本文の空き 本文領域と区別するために、図表と本文は 1 行程度は空きを設けて出力します。

図表の注釈 図表に注釈を付け加えるとき、注釈のサイズは本文よりも少し小さくします。

図表見出し 文書中のすべての図表に必ず見出し（図表見出し）を付けます。図には図見出しを、表には表見出しを付けます。見出しは図表と同じく中央揃えにします。場合によっては、図表見出しは本文に対して、書体とサイズを変更して出力する必要もあります。表見出しは表の上部、図見出しは図の下部に配置します。

通し番号 図表見出しには配置した順に一意の通し番号も表記します。これは「38 番目の図」という方法でも、「5 章の 6 番目の表」などでも構いません。通常、論文などの規模では章立てをする必要に迫られますので、図表見出しに付加する番号は「図 5.6」のように、章に連動して番号付けされます。

表罫線 欧文の表組みの場合、縦罫線は原則的に使いません。和文の場合でも、表に使用する罫線は最小にとどめることになります。

▼ 8.1.2 L^AT_EX での扱い

L^AT_EX では図表を浮動体 (float) と呼ばれる場所に一度退避させ、最適な位置に図表を配置しようと試みます (表 8.1). 浮動体として退避させた図表は少し制限の多い条件で組版されます.

表 (table) は tabular 環境で作成し、番号付けしたければ table 環境中に入れ子にします.

図 (figure) は picture 環境や画像ファイルを指定し、番号付けしたければ figure 環境中に入れ子にします. このようにするとそれらの図表は浮動体として扱われます. レポートや論文では図表に通し番号を付けるのは必須ですから、全ての表は table 環境の中へ、図は figure 環境の中に入れるのが良いでしょう.

図表を挿入するときに指定するのはその配置場所です. 基本的に L^AT_EX は図表をページの最上部か最下部に配置しようとして、それでも無理なときは別ページへと出力します. ユーザーはこれら図表 (浮動体) の配置場所を指定することが出来ます. 指定できる場所は表 8.2 となります. 位置指定は複数指定することが可能です. これらの位置指定は

表 8.1 浮動体の種類

	表	図
入れる環境	table 環境	figure 環境

表 8.2 浮動体の位置指定

記号	浮動体の配置する場所
h	まさにその場所に配置しようと試みます
t	ページ上部に配置しようと試みます
b	ページ下部に配置しようと試みます
p	浮動体を別ページに配置しようと試みます
!	無理やりその場所に配置します

table 環境や figure 環境の任意引数として渡します. figure 環境で例を示すと

```
\begin{figure}[htbp]
ここに図が入ります.
\end{figure}
```

のように使います.

図表用の見出しを出力するには \caption 命令を figure/table 環境中で使用します.

\caption{<図表見出し>}\label{<ラベル>}

前述のように表見出しは表の上部に出力するために、\caption 命令を先に、図見出しの場合は、図の後に \caption を先に記述します.

figure 環境中に表を入れたり、table 環境中に図を入れたりすることが出来ます. 他にも環境中に文字列を挿入することも可能です.

図表を文章中で参照するときは「上の図は何々」や「前述の図は何々」と参照してはいけません．必ず付加した通し番号で「図 3.8 は何々」として `\ref` 命令で参照します．そのためには `\label` 命令でラベルを付け加えることになります．間違っても手動で図表の番号を書かないで下さい．

8.2 表

L^AT_EX で表を作るために `tabbing` 環境, `tabular` 環境, `array` 環境の三つが用意されております．`array` 環境は 7.5.5 節にて紹介していますのでそちらを参照してください．`tabbing` 環境も簡単に表が作成できる環境なのですが, `tabular` のほうが記述が楽だと思いますので, ここでは `tabular` のみを紹介します．`tabular` 環境は次のように記述します．

```
\begin{tabular}{列指定子}
  a11 & ... & a1n & \\
  ⋮ & ⋮ & ⋮ & \\
  am1 & ... & amn & \\
\end{tabular}
```

行列とほぼ同じです．違うのは数式環境には入れなくても良いということです．

列指定子とはその `tabular` 環境における表の列数や縦方向の罫線などを決めるものです．`tablar` 環境で利用できる主な列指定子は表 8.3 の通りです．`tabular` 環境における

表 8.3 `tabular` 環境の主な列指定子

列指定子	意味
l	行列の縦 1 列を左揃えにする
c	行列の縦 1 列を中央揃えにする
r	行列の縦 1 列を右揃えにする
	縦の罫線を引く
	縦の 2 重罫線を引く
@{表現}	表現を 1 列追加する
p{長さ}	ある列の幅を直接指定する
*{回数}{項目}	回数分だけ項目を繰り返す

各要素 (成分) はアンド ‘&’ で区切ります．‘\\’ を行の終わりとしますので例えば 1 行 3 列の表は次のようになります．

```
\begin{tabular}{ccc}
  \LaTeX2.09 & \LaTeXe & \LaTeX3\\
\end{tabular}
```

L^AT_EX2.09 L^AT_EX 2_ε L^AT_EX3

横方向に罫線を引くには `\hline`, 要素の中で縦の罫線を引くときには `\vline` などを使います (表 8.4)．横方向の罫線を引くには `\hline` を, 行を連結するには `\multicolumn`

を使います .

表 8.4 tabular 環境中での罫線の命令

命令	意味
<code>\hline</code>	横に引けるだけの罫線を引く
<code>\hline\hline</code>	引けるだけの 2 重の横罫線を引く
<code>\vline</code>	要素の中で引けるだけの縦罫線を引く
<code>\cline{<範囲>}</code>	要素の罫線を行の範囲を指定して引く
<code>\multicolumn{<数値>}{<列指定子>}{<要素>}</code>	行を繋げて列指定子通りに要素を出力する

```
\begin{tabular}{|c|c|c|}
\hline
\multicolumn{3}{|c|}{\LaTeX} \\\hline
\LaTeX2.09 & \LaTeXe & \LaTeX3 \\\hline
\end{tabular}
```

L ^A T _E X		
L ^A T _E X2.09	L ^A T _E X 2 _ε	L ^A T _E X3

レポートや論文では表には表見出しを付けて中央揃えにするのが望ましいと思われます
ので以下のようなフォーマットになります .

```
\begin{table}[htbp]
\begin{center}
\caption{表の出力例}\label{tab:tabular:example}
\begin{tabular}{llcr}
\hline
出力例 & 1 & 2 & 3 \\\hline
\LaTeX の遷移& \LaTeX2.09 & {\LaTeXe}& \LaTeX3 \\\hline
\end{tabular}
\end{center}
\end{table}
```

上記のソースの出力例が表 8.5 となります . 毎回このような記述をしていたのでは疲れま

表 8.5 表の出力例

出力例	1	2	3
L ^A T _E X の遷移	L ^A T _E X2.09	L ^A T _E X 2 _ε	L ^A T _E X3

すので , 表用の mytab 環境を次のように定義します .

```
\newenvironment{mytab}[3][htbp]
{\begin{table}[#1]\begin{center}\caption{#2}\label{#3}}
{\end{center}\end{table}}
```

こう定義しておけば

```
\begin{mytab}[htbp]{中央揃えで見出しのある表の環境}{tab:hoge}
\begin{tabular}{lll}
```

```
\LaTeX2.09 & \LaTeXe & \LaTeX3\\
\end{tabular}
\end{mytab}
```

のように使うことができるわけです。

▼ 8.2.1 表中の脚注

tabular 環境中での脚注はうまく出力できないことが多いようです。その場合は `\footnotemark` と `\footnotetext` の二つを使います。

```
\footnotemark[<番号>]
\footnotetext[<番号>]{<注釈内容>}
```

`\footnotemark` で脚注記号を表示し、`\footnotetext` に注釈を書きます。

```
\begin{tabular}{|c|c|c|} \hline
  一つ目\footnotemark[1] &
  二つ目\footnotemark[2] &
  三つ目\footnotemark[3] \\ \hline
\end{tabular}
\footnotetext[1]{表中一つ目の脚注です。}
\footnotetext[2]{表中二つ目の脚注です。}
\footnotetext[3]{表中三つ目の脚注です。}
\\ ちょっと表示が変になっています。
```

一つ目 ^{*1}	二つ目 ^{*2}	三つ目 ^{*3}
-------------------	-------------------	-------------------

ちょっと表示が変になっています。

^a 表中一つ目の脚注です。
^b 表中二つ目の脚注です。
^c 表中三つ目の脚注です。

上記の方法ではうまくいかない場合は手動で脚注を付けることもできます。

```
\begin{tabular}{|c|c|c|}\hline
  一つ目${}^{\sim{a}}$ & 二つ目${}^{\sim{b}}$ &
  三つ目${}^{\sim{c}}$ \\ \hline
\end{tabular}
{\footnotesize \\
${}^{\sim{a}}$表中一つ目の脚注です。\\
${}^{\sim{b}}$表中二つ目の脚注です。\\
${}^{\sim{c}}$表中三つ目の脚注です。}
```

一つ目 ^a	二つ目 ^b	三つ目 ^c
------------------	------------------	------------------

^a 表中一つ目の脚注です。
^b 表中二つ目の脚注です。
^c 表中三つ目の脚注です。

▼ 8.2.2 表作成支援ツール

L^AT_EX で 0 から表を組むのは初心者には辛いかもしれません。GUI ベースのプログラムで表を作成し、それを L^AT_EX の tabular 環境の記述に変換するツールを使うと良いでしょう。Microsoft の Excel を使っている場合は中尾誠氏の Excel2latex

http://www32.ocn.ne.jp/~butcher_bird/Mac/Excel.html

や浦壁厚郎氏の Exel2tabular

<http://www.ne.jp/asahi/i/love/E2T/>

などがありますので参考にしてください。これらのプログラムは Microsoft の Excel で作成された表を \LaTeX のソースに変換します。

Microsoft の Excel ではなく OpenOffice.org の Calc を使っているならば阿部昌平氏の Calc2LaTeX

<http://web.hc.keio.ac.jp/~mr041754/calc2latex/indexj.html>

というものがあるそうです。これを使えば Calc で作成した表を tabular 環境に変換し、表として \LaTeX に貼り付けることが出来ます。

8.3 図

\LaTeX では画像に関する多くの処理をデバイスドライバに頼るしかないので、自分の使おうとしているデバイスドライバがどのような画像処理に対応しているのかを知っておきましょう。PDF を作成したいならば Dvipsd \LaTeX , PostScript ならば dvips を使うことになります。

図の挿入に関しての方法は大きく分けて 2 通りあります。一つはペイントソフトなどで書いた画像をそのまま取り込む方法、もう一つは \LaTeX の picture 環境で図を直接書く方法です。二つ目の方法の詳しいことは『 \LaTeX コンパニオン』[3]と『 \LaTeX グラフィックスコンパニオン』[4]を参照してください。一般的な図の取り込み方としては別のソフトウェアで Encapsulated PostScript (EPS) 形式で画像を保存し、それを graphicx パッケージで読み込みます。商用のソフトウェアであれば EPS 出力をサポートしています。Unix 系 OS の場合 \LaTeX と相性が良いと思われるのは Tgif や Ngraph でしょう。

▼ 8.3.1 EPS 画像の張り込み

EPS とは Encapsulated PostScript の略で Adobe 社の開発したページ記述言語です。PostScript にプレビュー用の情報を付加したファイル形式です。単一ページでベクトル画像を保存するのに良く使われます。EPS 形式の画像の場合は dvips を使うとファイル形式の変換を必要とせずにそのまま PostScript ファイルに取り込むこともできます。Dvipsd \LaTeX などは 1 度 Ghostscript の pdfwrite を使って EPS から PDF に変換した PDF ファイルを取り込みます。

\LaTeX での基本的な EPS 画像の扱い方を説明します。

1. 各アプリケーションで EPS 保存オプションをモノクローム 256 色 (8 ビット)、PostScript レベル 2 などに設定してから保存します。フォントはアウトライン化 (ベクトル化) できる場合は行います。カラー印刷する場合はカラーで構いませんが容量は大きくなります。
2. 文書のプリアンプルで graphicx パッケージを使うことを宣言します。
 $\backslash\text{usepackage}$ [デバイスドライバ]{graphicx}

PostScript 形式の文書を出力するならば、dvips を指定します。PDF を作成した

いときは *Dvipdfmx* を使うために *dvipdfm* を指定します．

3. 図を挿入したい場所に `\includegraphics` 命令を使ってファイル名を示します．

`\includegraphics[設定]{ファイル}`

仮引数には表 8.6 などのオプションが使えます．

表 8.6 *graphicx* パッケージで使える主なオプション

設定項目名	説明	値
<code>width=⟨幅⟩</code>	出力する図の幅を指定します	長さ, 100 mm など
<code>height=⟨高さ⟩</code>	出力する図の高さを指定します	長さ, 100 mm など
<code>angle=⟨角度⟩</code>	反時計回りの回転角度を指定します	$0 < \text{値} < 360$
<code>scale=⟨数値⟩</code>	図の拡大・縮小率を指定します	$0 < \text{値}$

例えば Ghostscript がインストールされている場所に `examples/` というディレクトリがあります．そこに `golfer.eps` という EPS 画像があります．この EPS 画像ファイル `golfer.eps` を \LaTeX の文書に張り込むには `golfer.eps` を \LaTeX の原稿のあるディレクトリにコピーして

```
\usepackage[dvips]{graphicx}% dvipdfm
```

をプリアンブルに書きます．そして `document` 環境の中で

```
\includegraphics[width=5cm]{golfer.eps}
```

とします．これを `latex` でタイプセットし、整形された DVI ファイル `⟨file⟩.dvi` をプレビューアで見ることができます．

上記の操作が問題なくできるでしょうか．基本的に \LaTeX で既存の画像を張り込むときは \LaTeX に標準で含まれている *graphicx* パッケージを使うことになると思われます．

画像の張り込みに関しては *graphicx* パッケージという統合的な方法が提供されています．このパッケージを使うことによって任意のデバイスドライバに合わせたコマンドを記述しなくても良いように工夫がされています．

まずこの *graphicx* を使うためにプリアンブルに以下の 1 行を追加します．

```
\usepackage{graphicx}
```

このとき使用するデバイスドライバが重要で最終的に PostScript 形式か、DVI 形式か、それとも PDF 形式が必要なのかによってオプションを変えます．Windows で印刷することも考えるならば最終的に PDF 形式にすると良いのかもしれませんが．そうするとデバイスドライバは *Dvipdfmx* を使うことになるでしょうから、*Dvipdfmx* の場合は

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

とします．*Dvipdfmx* の場合も *graphicx* においては *dvipdfm* と同じ設定になりますから *dvipdfmx* と書いてはいけません．

Unix 系 OS ならば PostScript のほうが良いでしょうから

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

とします。dvipsk であろうが pdvips だろうが dvips オプションを使います。他には xdvi や dviout も指定できます。dviout の場合は dviout がインストールされているフォルダの GRAPHIC/LATEX2E/dviout.def というファイルを \$TEXMF/tex/latex/graphics/ にコピーしてください。

既存の画像は基本的に \includegraphics 命令で読み込みます。

```
\includegraphics[<設定>]{<ファイル名>}
```

Windows の方で手持ちの画像のほとんどがビットマップで存在するならば dviout をデバイスドライバに選択すれば良いでしょう。EPS 画像が多いならば 1 度 EPS から PDF に変換してから Dvipdfmx を使うのが良いと思われます。Unix 系 OS ならば手持ちの画像を EPS に変換して dvips を使うことになるでしょう。

試しにご自分の持っている画像 (ファイル) を (デバイス) で取り込めるのかを試してみてください (行頭のパーセントは取り除き、images フォルダに ridt91.eps があると仮定します)。

```
%\documentclass{jarticle}
%\usepackage[dvipdfm]{graphicx}
%\begin{document}
\includegraphics[width=3cm]
  {images/ridt91}
%\end{document}
```



dviout の場合 EPS 画像を取り込むときは Ghostscript にて EPS を PDF に変換してから画像を表示しますから dviout の Ghostscript に関する設定を適切に行ってください。画像によってはページをはみ出したりしている場合があるでしょうし、表示が大きすぎる場合があるでしょう。その場合は取り込みに関する設定をします。

height=<高さ> 単位付きで画像の高さを指定します。

width=<幅> 単位付きで画像の幅を指定します。

scale=<数値> 画像の拡大率を指定します。

angle=<角度> 反時計回りに画像を回転する角度を指定します。

bb=<画像の位置情報> 画像のどの領域を使うべきかを指定します。'bb=0 0 640 480'

とすると原点を (0, 0) として縦横 '640 × 480' の領域を使うようにします。

noclip 画像用に使うべき領域を元の画像がはみ出している場合に画像を切り抜かないようにします。

clip 画像が確保された領域よりも大きい場合は切り抜きします。

draft 実際に画像を張り込まずに画像が占有するだろう領域を枠による代替表示になり、ファイル名も表示します。

レポートや論文などで図には図見出しを付けて中央揃えにするのが望ましいと思われるので

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=10cm]{images/file.eps}
\caption{図見出し}\label{fig:samplefig}
\end{center}
\end{figure}
```

のように使うことになります。ただし、これを毎回書くのは面倒なので次のような図用の `myfig` 命令を作成します。

```
\newcommand{myfig}[4][width=.8\textwidth]{%
\begin{figure}[htbp]%
\centering\includegraphics[#1]{#2}%
\caption{#3}\label{fig:#4}%
\end{figure}}
```

このように定義しておけば次のように使えます。

```
以上の考察から図~\ref{fig:sample}のような図が得られる。
\myfig[width=100pt,clip]{images/file.eps}{図の張り込みの例}{sample}
```

浮動体の図は DVI ファイルに出力されるときに思いもよらない場所まで旅をしますので、思い通りの場所に図を出力できなくても気にしないでください。そもそも図表に対して「上記の図は何々」とか「下記の図は何々」という表現は間違いで、全ての図表は「図 3.1 は何々」のように番号で参照します。ですから本来は図や表がどのような場所に旅立っても困らないはずです。

図などを反時計回りに 90° 回転させることがあるでしょう。その場合は `\rotatebox` 命令を使います。

`\rotatebox[〈設定〉]{〈角度〉}要素`

これは `\includegraphics` の任意引数に ‘angle’ を使ったことと同じです。`\rotatebox` は図に限らずあらゆる要素（表も可能）を回転します。〈設定〉の項目には以下のようなものがあります。

`origin=〈ラベル〉` 要素を回転するための原点を指定します。左 ‘l’、右 ‘r’、中央 ‘c’、上部 ‘t’、下部 ‘b’ が指定できます。

`x=〈長さ〉` x 方向の原点の位置を直接〈長さ〉を指定します。

`y=〈長さ〉` y 方向の原点の位置を直接〈長さ〉を指定します。

```
\rotatebox{70}{文字列など}
\rotatebox[origin=c]{60}{回転とか}
\rotatebox[origin=b]{50}{どう}
\rotatebox{30}{ですか?}
```

文字列など
 の
 回転とか
 は
 どうですか?

▼ 8.3.2 他のプログラムから張り込む方法その 1

例えば Excel や PowerPoint などのグラフや図表を張り込む場合大きく分けて次の 3 通りの方法があります。対象となるグラフや図表、ページは基本的に PS 形式の画像に変換することができます。この小節の話は Windows に限定します。

1. 該当のページなどをスクリーンショットでビットマップ画像として保存しその画像を EPS-Conv や ImageMagick で EPS 画像に変換してから張り込む。
2. ページなどの大きさを調整してから PostScript プリンターを通してプリンターファイルとして保存し、バウンディングボックスを eps2eps で調整してから張り込む。
3. ページなどの大きさを A4 の紙いっぱいにファイルとして印刷し、それを graphicx パッケージで回転と拡大・縮小をして張り込む。

一つ目と二つ目は手間がかかるうえに印刷品質も悪いのでここでは説明しません。三つ目の方法を解説します。

手順は以下の四つようになります。Microsoft の Excel のグラフを取り込む手順を例に出しています。

1. まず、Windows の[設定]で[プリンタの追加]をします。追加するプリンタは「ローカルプリンタ」で使用するポートは「ファイルへ出力」を選択し、ここでは「HP Color Laser Jet PS」というプリンタソフトウェアを選択します。テストページを印刷する場合は、出力ファイル名を「file.eps」のように拡張子を.eps にして保存します。正しく印刷されているかどうかを確認するには Ghostscript か GSView などで見ることができます。
2. 次に元となるグラフやページを作成します。このときモノクロ印刷を想定してグラフを調整します。印刷したいグラフを選択したならば、[ページ設定]で印刷の向きは「横」、用紙サイズは「A4」、[余白]タブで上下左右、ヘッダー、フッターの余白をすべて「0」に設定し、[ヘッダー、フッター]タブで何も出力しないことを選択し、[グラフ]タブで、印刷するグラフのサイズを「用紙サイズ」に合わせます。印刷品質は「白黒印刷」にするとファイルサイズも小さくなり処理速度も向上します。印刷プレビューで問題がないことを確認したならば、アプリケーション側での設定は終了です。
3. [ファイル]から[印刷]を選び、プリンタの名前を先程追加したプリンタに設定します。「OK」ボタンを押し絶対パスでファイル名を指定してファイルへ出力すれば、目的のグラフを EPS で保存することが出来ます。このとき保存するファイル名は半角英数字とし拡張子は.eps としてください。ファイルの保存先を原稿と同じ場所にすると問題も少ないでしょう。
4. 出力した filename.eps を L^AT_EX 文書中で参照します。graphicx パッケージを使って

```
\includegraphics[scale=0.4,angle=-90]{filename}
```

とすれば丁度良いあんばいで取り込めます．A4 の紙にグラフを出力した場合，0.4 倍の縮小率で，時計回りに 90 度回転させると \LaTeX 文書の中で丁度良いサイズとして張り込むことが出来ます．

以上の 2-4 の作業をグラフの数だけ繰り返すことにより， \LaTeX に対して Excel で作成されるグラフとほぼ同じイメージを挿入することが可能です．フォントがビットマップフォントでギザギザになるときはプリンタの設定で「TrueType フォントダウンロードオプション」のような項目があるはずですので，これを「アウトライン」にすると良いでしょう．

▼ 8.3.3 他のプログラムから取り込むとき

Mathematica や Illustrator からグラフや画像を取り込むときには幾つかコツが必要です．112 ページ 8.3.2 節 での Excel での張り込み方が他のアプリケーションでも適用できる場合が多いので，上記の方法を試してみてください．

どのプログラムを使用しても最終的に出力したい画像のサイズを元のプログラム側で調節してから \LaTeX に張り込むようにすると問題も少ないでしょう．graphicx パッケージの拡大縮小を使うと印刷品質が落ちます．各プログラムにおける設定方法は以下の通りです．

Illustrator まず文字はアウトライン化します．ツールバーの[別名で保存]でファイル形式を 'Illustrator EPS' として保存します．EPS 形式での保存オプションで「サムネールを作成」のチェックを外して，「フォントデータを含む」のチェックをはずしてください．ポストスクリプトのレベルは 'PostScript level 2' としてください．プレビューについては '8-bit IBM PC' で良いと思います．Illustrator の場合はサイズが自動的に調節されますので用紙サイズを設定する必要はありません．誤ってサムネールを付けた場合バージョンによってエラーになるので Tomas Rokicki 氏の fixill.pl を使うと良いでしょう．Perl スクリプトで書かれており，実行するためには Perl が必須で

```
$ fixill <input.eps >output.eps
```

のように使います．

photoshop [ファイル]，[複製を保存]を選び「保存形式」を 'Photoshop EPS' にして保存する．保存オプションで「エンコーディング」は 'ASCII' とする．ビットマップ画像はエンコーディングで圧縮しないほうが印刷品質が良いでしょう．

Mathematica ツールバーから[ファイル]の[特殊な形式で保存]を選び \TeX(X) を選びます．そうすると数式やグラフなどが自動的に $\text{\LaTeX 2_{\epsilon}}$ 形式に保存されます．またグラフは EPS 形式で filename.eps という名前で保存されます．Mathematica の場合出力される EPS 画像のバウンディングボックスが正常に出力されないことがあるので \LaTeX で正しく処理できない場合があります．出力された

filename.eps というファイルをテキストエディタで開けば

```
%%BoundingBox: 91.5625 3.1875 321.938 190
```

のような記述があります。これは画像を平面上のどこに配置するかを指定するもので、左から 2 次元平面上の始点の x_0 と y_0 、終点の x と y に対応します。また、通常はこの値は整数値が推奨されます。上記の数値を四捨五入して整数に直して取り込んでください。

MATLAB グラフを表示している MATLAB プログラムのウィンドウのツールバーにある[ファイル]から[エクスポート]を選び、ファイルの種類を‘EPS Level 2’にし、任意の名前をつけて保存します。Illustrator 形式での出力もサポートされていますので、お持ちの場合はグラフを編集できます。

▼ 8.3.4 図を二つ横に並べる

2 段組の場合はそのようなことはありませんが、1 段組の場合は一つの図だけでは両脇が開いてしまうのでそこに二つの図を‘(a)’と‘(b)’として挿入したいときがあります。このようなときは minipage 環境を使います。以下のように入力する例もあります。

```
\begin{figure}[htbp]
  \begin{minipage}{.47\textwidth}
    \centering% ここに図 (a) を入れる
    (a) 初期値 $c=0.6$ 
  \end{minipage}
  \hfill
  \begin{minipage}{.47\textwidth}
    \centering% ここに図 (b) を入れる
    (b) 初期値 $c=1.0$ 
  \end{minipage}
  \caption{1 段組で横に図を二つ並べる}
\end{figure}
```

両方の図の番号を別にしたいときも同様に記述します。

▼ 8.3.5 EPS 以外の画像の張り込み

L^AT_EX では EPS 以外の画像の張り込みも可能ですが少々癖があります。お手元には GIF や BMP, JPEG などのビットマップ画像があると思います。ビットマップ画像を EPS に変換できるツールは無料でたくさんあり、有名なものが ImageMagick です。Windows 専用ですが EPS-Conv

<http://hp.vector.co.jp/authors/VA023018/>

というソフトウェアを導入することにより、現存するほとんどの画像を EPS に変換可能です。これらのアプリケーションはあくまでビットマップを EPS 形式で保存するだけなのでファイルサイズが非常に大きくなります。

デバイスドライバ (5.4 節参照) として Dvipdfmx を使うと EPS 形式以外の画像でも取り込むことができます。変換しなくても取り込める形式は PDF, JPEG, PNG, MetaPost の 4 種類です。Windows の BMP などは低圧縮の JPEG などに変換します。image.png があった場合端末などで

```
$ ebb image.png
```

とすれば image.png 用の image.bb が作成されます。ebb は dvipdfm に同封されています。この image.bb は画像ファイルの縦横を正しく扱うためのファイルです。image.bb を見れば分かりますが、中身は

```
%%Title: ./image.png
%%Creator: ebb Version 0.5.2
%%BoundingBox: 0 0 595 841
```

となっています。‘BoundingBox’ とは原点座標と画像の縦横の長さの値です。次にソースファイルを以下のようにします。

```
\documentclass{jsarticle}
\usepackage[dvipdfm]{graphicx}
\begin{document}
\includegraphics[width=3cm]{image.png}
\end{document}
```

後はいつも通りにタイプセットして DVI ファイルを生成し Dvipdfmx で PDF を作成すれば良いことになります。

8.4 描画の方法

L^AT_EX で図を取り扱う手段はいくつも存在します。写真のような画像を graphicx パッケージなどを使って張り込む方法と、1 から描画する方法です。graphicx パッケージを用いて既存の画像を張り込む方法は 8.3 節を参照してください。画像をまだ作成していない段階での描画の方法を紹介します。

描画の方法は大きく分けて二つあります。一つは L^AT_EX 自身の能力で描画する方法と \special 命令を使い他のプログラムへ描画をゆだねる方法です。一般に L^AT_EX における描画の能力は T_EX 譲りのシステムのお陰で貧弱なものとなっています。簡単な図を作成するならば L^AT_EX に備わっている picture 環境による描画を行うのが手軽です。

▼ 8.4.1 化学式・化学構造式

化学式や化学構造式を描くための藤田眞作氏による X^{き む て ふ}M_TE_X パッケージを使うと良いでしょう。これは L^AT_EX の picture 環境と epic を使ってベンゼン環やその他多くの化学式を描くことができます。X^{き む て ふ}M_TE_X について詳しく知りたい方は藤田眞作氏の書いた X^{き む て ふ}M_TE_X [8] を参照してください。

▼ 8.4.2 グラフの描画

L^AT_EX にグラフを挿入するには様々な方法があります．Windows の方であれば Excel で作成したグラフを EPS で保存し，それを graphicx パッケージで読み込むという方法（113 ページ 8.3.3 節参照）があります．巷の表計算ソフトなんて使いたくない方は Thomas Williams 氏と Colin Kelley 氏らによる Gnuplot を使うと良いでしょう．gnuplot はバージョン 3.7 に関しては山賀正人氏が，バージョン 3.8 に関しては尾田晃氏がプログラムの日本語化をされています．また gnuplot のマニュアルに関しても竹野茂治氏らによって行われています．

<http://takeno.iee.niit.ac.jp/~foo/gp-jman/>

制御系では SciLab というのがあります．マニュアルが大野修一氏らによって日本語化されています．

<http://www.ecl.sys.hiroshima-u.ac.jp/scilab/docjp.html>

John Eaton 氏らによる Octave というのもありますので調べてみてください．

第 9 章

L^AT_EX の応用

以下に示すコマンドなどはレポート・論文作成には必要不可欠という程の要素ではありませんので、このような機能もあるという程度でご覧ください。

9.1 レイアウトの制御

L^AT_EX ではユーザーが意図的に改行や改ページを行わなくても良いように工夫されています。どうしても自分の思い通りにページをレイアウトできないときは強制的なレイアウト命令を使います。ページ区切りを制御したいならば

`\newpage` 改ページします。2 段組の場合は次の段までの改ページになります。

`\clearpage` 未出力の浮動体を配置してから改ページします。2 段組の場合は本当の次のページまで改ページされます。

`\cleardoublepage` 次のページが奇数ページになるように改ページします。これを奇数起こしとか改丁と呼びます。

の四つの命令が使えます。

空白を制御するには以下の四つの命令が使えます。

`\hspace{<長さ>}` 長さ分の横方向の空白を挿入します。行頭では有効ではありません。

`\hspace*{<長さ>}` 行頭でも横方向の空白を挿入します。

`\vspace{<長さ>}` 長さ分の縦方向の空白を挿入します。ページの先頭・末尾では有効ではありません。

`\vspace*{<長さ>}` ページの先頭・末尾でも縦方向の空白を挿入します。

`\hspace{1cm}` 空白制御用のコマンドは行頭では意図的に `\vspace{0.5cm}` アスタリスクを付けます。 `\par`

`\hspace{1cm}` 段落の途中に縦方向 `\hspace{1cm}` の空白を挿入すると、段が改行されてから縦に空白が挿入されます。

空白制御用のコマンドは行頭では意図的にアスタリスクを付けます。

段落の途中に縦方向の空白を挿入すると、段が改行されてから縦に空白が挿入されます。

9.2 あらかじめ定義されている見出しの変更

「目次」や「参考文献」などの見出しは `\tableofcontents` 命令や `thebibliography` 環境によって出力されます．この見出しの文字を変更するには次のようにします．

```
\renewcommand{\refname}{関連書籍}
```

標準的な和文の文書クラスでは表 9.1 の見出しが定義されています．`\bibname` 命令は

表 9.1 定義済みの見出しの変更

命令	意味	標準的な定義
<code>\prepartname</code>	部見出し番号の前の文字	第
<code>\postpartname</code>	部見出し番号の後の文字	部
<code>\prechaptername</code>	章見出し番号の前の文字	第
<code>\postchaptername</code>	章見出し番号の後の文字	章
<code>\contentsname</code>	目次の見出しの	目次
<code>\listfigurename</code>	図目次の見出し	図目次
<code>\listtablename</code>	表目次の見出し	表目次
<code>\bibname</code>	<code>thebibliography</code> 環境の見出し	参考文献
<code>\figurename</code>	図見出し番号の前の文字	図
<code>\tablename</code>	表見出し番号の前の文字	表
<code>\appendixname</code>	<code>appendix</code> 環境での見出しの前の文字	付録

`jreport` や `jbook` などでの定義で `(j)article` では `\refname` となっています．奥村晴彦氏の `jclasses` では節見出し番号の前と後にも文字列を表示できるようになっています．

```
\renewcommand{\presectionname}{第}
\renewcommand{\postsectionname}{節}
```

のように `\presectionname` や `\postsectionname` を再定義します．

9.3 多段組

L^AT_EX では通常 1 段組と 2 段組しか制御できません．

```
\onecolumn
\twocolumn[要素]
\columnsep    (2 段組のときの段間)
\columnseprule (2 段組のときの段間に引く罫線の太さ)
```

1 段組みにするためには `\onecolumn` を使い，2 段組にするには `\twocolumn` を使います．`\twocolumn` は改ページをしてから 2 段組を作成しようとします．そのため任意引数に何らかの要素を与えるとその要素をページ上部に 1 段組で出力します．

```
\columnsep 2zw
\columnseprule .4pt
\twocolumn[{\large\LaTeXe はどうです?}]
ここからの文章が 2 段組になるでしょう。{\LaTeX}での
多段組の実現は難しいそうです。
```

2 段組にすると図表は用紙の文章幅 `\textwidth` ではなく 1 段分の幅 `\columnwidth` で張り込むことになります。また以下の二つの環境が使えます。

```
table*環境
figure*環境
```

`table` 環境や `figure` 環境にアスタリスクを付けるとその環境を 1 段分の幅でページの下部か上部に配置しようとします。

`\twocolumn` を使って 2 段組をすると最終ページの段の高さが揃わないので、格好悪いでしょう。これは `multicol` パッケージで 2 段組にすると段が揃いますし、`balance` パッケージを使っても可能です。

9.4 箱の操作

まずは \LaTeX で用意されている箱について説明します。これらは `ltboxes.dtx` で定義されています。 \LaTeX における箱というのは文章や段落、数式や図表などの要素を格納する領域のようなものです。 \LaTeX の箱には高さ、幅と深さの 3 種類の長さを持っています。さらに箱のどの点を基準にするかという基準点という座標も持ち合わせています。

▼ 9.4.1 枠のない箱

\LaTeX ではなんと簡単に複数の要素を一つの箱に収めることができます。

```
\makebox[幅][位置]{要素}
```

`\makebox` では箱の幅と箱の中の要素の位置を指定できます。箱の幅よりも要素の幅が狭いときに箱の左側に配置 ‘l’、中央に配置する ‘c’、右側に配置する ‘r’、最後に要素を均一に配置する ‘s’ の四つを使うことができます。

```
\makebox[3zw][l]{未来}と
\makebox[3zw][c]{函館}と
\makebox[5zw][r]{北海道}と
\makebox[5zw][s]{G o o d !}です。
未来 と 函館 と 北海道と G o o d !です。
```

要素の幅分の箱を作りたければ `\mbox` を使います。

```
\mbox{要素}
```

引数を省略すると要素分の幅を確保し `\makebox` を使うよりも効率が良いです。

```
\hspace*{\fill} 単なる予想ですが、この箱の
中では恐らく\mbox{改行が起こりません。}
単なる予想ですが、この箱の中では恐らく
改行が起こりません。
```

▼ 9.4.2 枠のある箱

複数の要素を一つの塊として扱うようにするのが L^AT_EX における箱の役割のようなものです。箱には枠を付けることもできます。

```
\framebox[幅][位置]{要素}
```

`\framebox` も `\makebox` とほぼ同じですが罫線の太さ `\fboxrule` と罫線と要素の間隔 `\fboxsep` の二つの長さを設定できます。`\fboxrule` は罫線の太さを、`\fboxsep` は枠と要素との距離を長さで指定します。

```
\framebox[3zw][l]{未来}と
{\fboxrule=3pt\framebox[3zw][c]{函館}}と
\framebox[5zw][r]{北海道}と
\framebox[5zw][s]{G o o d !}です。
```

未来 と 函館 と 北海道 と G o o d ! です。

`\makebox` と同じように引数を省略すると要素分の幅を確保する `\fbox` が使えます。

```
\fbox{要素}
```

これは `\fboxsep=0pt\fbox{びったりです}`。
 こちらは `\fboxrule=.8pt\fbox{若干太い}`。

これは びったりです。こちらは 若干太い。

▼ 9.4.3 広範囲な箱

指定した箱の大きさで段落を組む `\parbox` 命令もあります。標準では字下げがされませんので必要があれば `\parindent` に長さを代入してください。

```
\parbox[位置][高さ][要素の位置]{幅}{文字列}
```

`\parbox` で作成された箱の基準をどこにするのかを `<位置>` で指定します。位置には上部 `'t'`、中央 `'c'`、下部 `'b'` の三つが使えます。標準では中央になります。

```
\parbox{13zw}{段落が終わる命令\par
を使っても改行されますが\par
標準では字下げされません。}
```

段落が終わる命令
 を使っても改行されますが
 標準では字下げされません。

```
\parbox[c]{4zw}{箱が中央に。}\ldots
\parbox[t][3zw][c]{4zw}{文字が中央で
上が基準}\ldots
\parbox[b][3zw][t]{4zw}{文字が下で下
が基準か}\ldots
\parbox{3zw}{文字が上で下を基準}
```

箱が中央
 に。 … 文字が中
 央で上が
 基準

文字が下
 で下が基
 準か … 文字が
 上で下
 を基準

ページのような箱を組む `minipage` 環境もあります。

```
\begin{minipage}[<位置>]{<幅>}
ページ内容
\end{minipage}
```

minipage 環境では段落が組めますし、脚注の出力も可能です。

この環境は~

```
\begin{minipage}[t]{7zw}
ページを組みあげるので脚注\footnote{脚注で
す。}
もページの中に出力されます。
\end{minipage}
~のようになります。
```

この環境は ページを組み のようになります。

あげるので脚
注^aもページの
中に出力されま
す。

^a 脚注です。

▼ 9.4.4 罫線と下線

箱とは違うのですが罫線をここで紹介しておきます。

```
\rule[<上げ率>]{<幅>}{<高さ>}
```

\rule 命令は使いものになります。見えない罫線を引くこともできます。例えば幅が 0pt でも高さのある罫線、高さが 0pt でも幅のある罫線が使えますから、こんな使い方もできるわけです。枠の見える状態での例を見てください。

```
未来\fbbox{\rule{0pt}{3zw}\rule{4zw}{0pt}}
函館\fbbox{\rule{0pt}{3zw}\rule{2zw}{0pt}}
```

未来



函館

箱とは違うのですが下線も紹介しておきます。下線は \underline を使います。

```
\underline{<要素>}
```

\underline の中に箱を入れることもできますし、何を入れても構いません。

\underline{\fbbox{枠付きの箱}}の下線はこのようにしますし、もちろん\underline{下線}も表示できます。

枠付きの箱の下線はこのようにしますし、もちろん下線も表示できます。

9.5 空白の挿入

L^AT_EX にはいろいろ空白が用意されているのですが、それらは空気に含まれます。単語間に挿入される程度の空きを基準とするとその 4 倍の空きを ‘quad’ (クワタ) と呼びます。和文組版では空きの基準となるのは全角 1 文字分の幅であり、これを全角空白などと呼びます。全角空白一つ分の空きを全角空き、全角空白二つ分の空きを倍角空きと呼びます。さらに 4 分の 1 の場合は四分空き、6 分の 5 ならば二分三分と呼んだりします。欧文の ‘quad’ と和文の「クワタ」では若干長さが異なりますので、本冊子では二つを区別して表します。

▼ 9.5.1 水平方向の空き

水平方向の空きにはその両側での改行を許すものと許さないものがあります．主な空きを制御する命令は表 9.2 の通りです．表 9.2 は基本的に空きの前後での改行を行っても良いことになっています．

表 9.2 改行を許す水平方向の空き

命令	意味
<code>_</code>	適切な単語間空白（約 $1/4$ quad 分）
<code>\quad</code>	1 quad 分の空き
<code>\qquad</code>	2 quad 分の空き
<code>\enspace</code>	$1/2$ quad 分の空き
<code>\enskip</code>	適切な約 $1/2$ quad 分の空き
<code>\thinspace</code>	$1/5$ quad 分の空き
<code>\negthinspace</code>	$-1/5$ quad 分の空き

ユーザが`{\quad}`原稿の中`{\qquad}`で空きの調節を直接\するのは好ましくない．

ユーザが 原稿の中 で空きの調節を直接 するのは好ましくない．

表 9.2 の命令は改行を許しますが表 9.3 では空きの前後での改行を許しません．改行を許さないので行頭・行末が不揃いになるときがあります．

表 9.3 改行を許さない水平方向の空き

命令	意味
<code>\,</code>	$3/18$ quad 分の空き
<code>\:</code>	$4/18$ quad 分の空き
<code>\;</code>	$5/18$ quad 分の空き
<code>~</code>	適切な単語間空白

Donald~E. Knuth made \TeX\@.
Leslie~Lamport made \LaTeX\@.

Donald E. Knuth made T_EX. Leslie Lamport
made L^AT_EX.

自分で水平方向の空きの長さを指定するならば `\hspace*` 命令が使えます．

`\hspace*{[長さ]}`

アスタリスクをつけると行頭・行末でも使えるようになります．奥村晴彦氏の `jsclasses` を使っているときには `'pt'` や `'cm'` などの単位は使わずに `'truept'` や `'truecm'` などを使わないと長さがずれます．これが面倒ならば文章で使われているフォントに応じて基準の変わる `'em'` や `'zw'` などを使ってください．

▼ 9.5.2 垂直方向の空き

自分で長さを指定する垂直方向の空きにおいては`\addvspace`と`\vspace*`の二つが使えます。`\vspace*`はアスタリスクを付けないとページの最上部・最下部では有効になりません。あらかじめ長さの決まっている垂直方向の空きとして`\smallskip`、`\medskip`、`\bigskip`の三つがありますが、これはスキップと呼ばれるもので可変長の空きが挿入されます。「大体で良いからこれくらいの空きを入れてね。」程度の意味を持っています。垂

表 9.4 垂直方向の空き

命令	意味
<code>\smallskip</code>	3 pt ± 1 pt の空き
<code>\medskip</code>	6 pt ± 2 pt の空き
<code>\bigskip</code>	12 pt (+4 pt か -2 pt) の空き

直方向の空きは紙面の多くの部分を空きで占有するので無駄が多くなります。L^AT_EX では図表と段落のあいだやそのほか必要と思われるところには半自動的に空きが挿入されるようになっておりますので、闇雲に垂直方向の空きを挿入するのは好ましくないと思われます。

長さを自分で指定して空きを挿入する場合は`\vspace*`と`\addvspace`が使えます。

```
\addvspace{<長さ>}
\vspace*{<長さ>}
```

`\vspace*`のアスタリスクを外すとページの最上部・最下部での空きの挿入が有効になりません。`\addvspace`は直前の空きがどれくらいかも調べているので`\vspace`よりも適当な空きを挿入します。

この`\vspace*{2zw}`だと全角 2 文字分の垂直方向の空きが挿入されるところと思われます。

このだと全角 2 文字分の垂直方向の空きが挿入されるところと思われます。

9.6 原稿を複数のファイルに分ける

大規模な文書になるとそれを一つのファイルにまとめるのは効率が悪い場合があります。第 3 章は田中さんが編集し第 5 章は斉藤さんにお任せする、という状況では第 3 章と第 5 章の原稿は別々に存在させたいものです。この場合は原稿を複数のファイルに分けます。

```
\include{<ファイル名,...>}
\input{<ファイル名,...>}
\includeonly{<ファイル名,...>}
```

`\include` 命令はファイルを読み込むときに必ず新しいページから始めます。大規模な文書で章の区切りや節の区切りなどで使用します。この命令で取り込むときはファイルを

章ごとに (`\chapter` ごと) に分けることが考えられます。 `\input` はそのままの意味で指定されたファイルをそのまま親の L^AT_EX のソースファイルに取り込みます。取り込むファイルの拡張子が `.tex` ならば拡張子を省略しても構いません。

例えば論文を作成する場合は次のように分割することも出来ます。

```
\documentclass[dvipdfm,papersize]{jsbook}% クラスファイル
\usepackage{funcs}% スタイルファイル
\usepackage{amsmath,amssymb,bm,verbatim,listings}% 必要なマクロ
\begin{document}
\frontmatter%      前付
\include{0preface}% 前書き
\include{1thanx}%   謝辞
\mainmatter%       本文
\include{2joron}%   序論
\include{3honron}%  本論
\include{4keturon}% 結論
\begin{appendix}%   付録
\include{5code}%     付録: ソースコード
\end{appendix}%
\backmatter%        後付
\bibliographystyle{jplain}% 文献形式
\bibliography{ron}%  参考文献
\end{document}
```

9.7 用語の統一

大規模な文書の場合は、用語の統一というのが必要になってきます。一つの文書を複数の訳者で共同翻訳するときに専門用語の場合や新語の場合は語句を統一しなければ、読者を混乱させます。統一されていない事態を避けるためにはマクロを作成しておきます。

```
Hello, everyone! I'm a student at Future University-Hakodate.
```

という文章があったとして ‘Future University-Hakodate’ という用語が新語であったとしましょう。この用語をどんな単語に訳すのかをまだ決められない段階では次のようなマクロを作成します。

```
\newcommand*{\FUN}{Future University-Hakodate}
```

訳者のあいだで用語の訳が決まったならば

```
\newcommand*{\FUN}{公立はこだて未来大学}
```

とします。他にも人名や専門用語などで、非常に長い文字列を文書の中で何度も記述しなければならないときは、上記のように `\newcommand*` 命令で文字列を定義することになります。

ただし、欧文の場合は `xspace` を使わなければ、適切な空白が挿入されない場合がありますので、次のように定義します。

```
\newcommand*{\<<文字列>>\xspace}
```

```
%\usepackage{xspace}
```

```
\newcommand*{\LC{Logical OR}}% ( × )
```

```
\LC is good? ‘\LC’ is also ok.
```

```
\renewcommand*{\LC{Logical OR\xspace}}
```

```
\LC is funky! ‘\LC’ is also ok.
```

Logical OR is good? ‘Logical OR’ is also ok.

Logical OR is funky! ‘Logical OR’ is also ok.

9.8 URL の記述

近年は公式な文書の中にもウェブ上への参照先を示すために URL と呼ばれるアドレスを書く場合があります。これを \LaTeX で実現しようと思えば `\verb` 命令が使えと思うのですが脚注の中では使えない、引数の中で使えないという事態に陥ります。このようなときは Donald Arseneau 氏による `url` を使うと良いでしょう。使い方は `\verb` 命令とほぼ同じで ‘%’ や ‘#’ などの特殊記号に対して特別な対処をしなくともそのまま記述できます。URL に対しては `\url` を、パスやファイルを示す場合は `\path` を使います。e-mail などを表記する場合は新規に `\email` 命令を定義します。

```
\newcommand{\email}{\begingroup \urlstyle{rm}\Url}
```

使われるフォントは `\urlstyle` で指定します。スラッシュやピリオドの位置などで自動的に改行されます。

```
\newcommand*\email{\begingroup
```

```
  \urlstyle{rm}\Url}
```

```
\newcommand*\dir{\begingroup
```

```
  \urlstyle{tt}\Url}
```

```
\url{http://www.server.com/dir/file.htm}
```

```
にアクセスしたら\email{name@server.ac.jp}
```

```
というメールアドレスがあったので、
```

```
\dir{/usr/local/bin/octave}を削除した。
```

<http://www.server.com/dir/file.htm> にアクセスしたら `name@server.ac.jp` というメールアドレスがあったので、`/usr/local/bin/octave` を削除した。

9.9 原稿の執筆支援

\LaTeX の原稿を入力するときにそれを支援する環境がいくつかあります。

▼ 9.9.1 入力支援統合環境 YaTeX

有名なプログラムとして GNU Emacs 上で動作する広瀬雄二氏の作成された YaTeX が便利です。詳しいことは YaTeX の公式ページ

<http://www.yatex.org/>

から情報を集めてください。

詳細は広瀬氏の書かれたマニュアルがありますのでそちらを参照してください。

同じように GNU Emacs 上で動作する AUCT_EX というものもあります。詳細はご自分で調べてみてください。

▼ 9.9.2 入力支援統合環境 WinShell

Ingo H. de Boer 氏らが作成した WinShell は Windows 上で動作する L^AT_EX の統合支援環境です。

<http://www.winshell.de/>

日本語化やその他周辺情報は奥村晴彦氏のウェブページを参考にしてください。

<http://oku.edu.mie-u.ac.jp/~okumura/texwiki/>

▼ 9.9.3 latexmk

Make はちょっと大げさなのでもう少し簡単に実行できる John Collins 氏による latexmk を紹介しておきます。Perl スクリプトで書かれていますので、導入する前に Perl をインストールします。まず latexmk を動かすために latexmk.pl に変更を加えます。ただ単に欧文用のプログラムを和文対応にするくらいです。

```
$latex = 'latex'; → $latex = 'platex';
$bibtex = 'bibtex'; → $bibtex = 'jbibtex';
$makeindex = 'makeindex'; → $makeindex = 'mendex';
$dvips = 'dvips'; → $dvips = 'dvipsk';
```

あとはターミナルから

```
$ latexmk file
```

とすると標準では DVI ファイルが作成されます。Bib_TE_X や MakeIndex を使っていても自動的にそれらの処理が行われます。ただし

```
\bibliography{biblio.bib}
\include{file1.tex}
```

とするよりは拡張子を省略して

```
\bibliography{biblio}
\include{file1}
```

とすることになります。

第 10 章

論文のサンプル

今まで様々な情報を提供してきましたが、実際に自分で論文の書式を書き起こすのは大変かもしれません。そこでこの章では卒業研究などで提出する概要レポート、いわゆる中間報告と卒業研究で最終的に提出する卒業論文の例を示します。

学位論文などの書式である文書クラスは大学や学会などから指定されます。当大学の場合は `funthesis.cls` というファイル名で卒業論文のウェブページにて配布されているものと思います。学会なども同様に独自のクラスファイルを配布していますので、その書式に合わせて書きます。

10

10.1 中間報告のサンプル

中間報告は当大学の規定で、2 ページ程度にまとめることになっています。この場合、題名、概要、参考文献、図表などを要領よく整理することが重要になります。そのため中間報告では 2 段組にするのが良いでしょう。2 段組にすると以下のような利点があります。

- 1 段組よりも適切な文字数で改行される。
- 図を取り込むときに `\columnwidth` を使える。

中間報告のサンプルソースファイルと出力結果をご覧ください。このサンプルに使っている文書クラスは奥村晴彦氏の `jsarticle` です。サンプルのソースファイル中には注意事項なども書いていますので参考にしてください。

`jsarticle` を使わずに `article` や `jarticle` を使わなければならないならば、概要については表題の下に 1 段組で出力するでしょうから `abstract` パッケージを使ってみてください。`abstract` では `\twocolumn` 命令の任意引数の中で `\onecolabstract` 命令を使います。

```
\twocolumn[{\maketitle
\begin{onecolabstract}
概要部分
\end{onecolabstract}}]
```

`jsarticle` を使った例のソースファイルです。

```

\documentclass[dvipdfm,twocolumn]{jsarticle}
\columnseprule 0.5pt% 段間の罫線
\usepackage{epic,eepic,amssymb,amsmath,graphicx,url}
\title{2 段組での中間報告のサンプル}
\author{{\small システム情報科学部 情報アーキテクチャ学科}\small
m1202147 渡辺 徹 \small 指導教官 未来 太郎}
\date{\today} % \today 命令は文書を作成した日付が代入される
% 本文開始
\begin{document}
\begin{abstract}% 概要
論文作成においては\LaTeX{}を使用するのが望ましいが、近年では事務処理用の
Word がその代わりとなっているように見受けられる。今回は、はこだて未来大学
においてどの程度 Word や\LaTeX{}が浸透しているのかを 2003 年度の卒業研究から
提出される中間レポートを参考に統計を取ってみた。結果は予想通り Word 人口が
圧倒的に多かった。また、この中間報告のサンプルの内容は出たら目であるので、
あくまで入力例として参考にしてもらいたい。
\end{abstract}
\maketitle% 表題

\section{目的}
当大学では卒業研究の中間報告として中間レポートを提出するようになってい
る。各自がどのようなアプリケーションを使っているのかを調査することが今
回の目的である。

\section{方法}
直接研究生にアンケートをとったわけではなく、ウェブページ上で 2003 年 9 月
10 日までに提出されているレポートを調査対象とした。

\section{結果}
提出されているレポートを大まかに調査した結果が表~\ref{2bansenji}となる。
これは研究生がどのようなアプリケーションで中間レポートを作成したのかを
調べた結果である。どうしても判別できないものは\emph{その他}の項目に入れてあ
る。レポートの最終形態ではなく、原稿を作成する段階で使ったアプリケーシ
ョンを示している。
\begin{table}[htbp]
\begin{center}
\caption{データの分析結果}\label{2bansenji}
\begin{tabular}{lrrr}
\hline
項目 & & 人数 (人) & 割合 (\%) \\
\hline
Word & 75 & 45.2 \\
\LaTeX{} & 26 & 15.6 \\
HTML & 54 & 32.5 \\
Illustrator & 4 & 2.4 \\
OpenOffice & 1 & 0.6 \\
その他 & 6 & 3.0 \\
合計 & 166 & 100
\end{tabular}


```

```
\end{tabular}
\end{center}
\end{table}
```

これらの結果は二次的に入手した情報のため、データに若干の誤りがある。直接アンケートをとって調べればもっと正確な情報が収集できるが、今回は簡易的な形をとった。

```
\section{考察}
```

以上の結果から、現在 HTML で作成している人物は Word を使う事になるだろう。結果があくまで中間報告である事を考えれば、Word 人口がこれから増えることは明白である。今度の働きかけ次第で当大学の \LaTeX 人口を増加させることも可能である。

この現象を天下りのフーリエ変換で解析する。まず、フーリエ変換で関数 $f(x)$ を定義する。この関数 $f(x)$ は変換のための区間を必要とするので、区間を $[-L, L]$ とする。すると以下の式が定義から導出される。

```
\begin{eqnarray*}
f(x) &= & \frac{a_0}{2} + \sum_{n=1}^{\infty} \left( a_n \cos \right. \\
&& \left. \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \\
a_n &= & \frac{1}{L} \int_{-L}^L f(u) \cos \frac{n\pi u}{L} du \\
b_n &= & \frac{1}{L} \int_{-L}^L f(u) \sin \frac{n\pi u}{L} du
\end{eqnarray*}
```

よって、次式^(\ref{eq:fourier1})が新たに得られる。

```
\begin{eqnarray}
f(x) &= & \frac{1}{2L} \int_{-L}^L f(u) du \nonumber \\
&& + \sum_{n=1}^{\infty} \left[ \frac{1}{L} \int_{-L}^L f(u) \cos \frac{n\pi x}{L} du \cdot \cos \frac{n\pi x}{L} \right. \\
&& \left. \right. \text{right.} \nonumber \\
&& + \frac{1}{L} \int_{-L}^L f(u) \sin \frac{n\pi u}{L} du \cdot \sin \frac{n\pi x}{L} \left. \right] \\
&& \text{\label{eq:fourier1}}
\end{eqnarray}
```

式^(\ref{eq:fourier1})を $(L \rightarrow \infty)$ にしたりしてフーリエ変換は一般に式^(\ref{eq:fourier2})のように書き表すことができる。

```
\begin{equation}
F(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(u) e^{-t\alpha u} du \text{\label{eq:fourier2}}
\end{equation}
```

式^(\ref{eq:fourier2})を使って今回の結果を解析することは、現段階では非常に困難であると容易に考察できる。

```
\section{今後の展望}
```

今回得られた調査結果を下に Gnuplot でデータをプロットする作業が続くものと思われる。また、グラフは主に Gnuplot から挿入するのが望ましいとされる。Gnuplot から挿入したグラフは図^(\ref{fig:sample})となる。

```
\begin{figure}[htbp]
\begin{center}
%\input{abstgnu.tex} % ファイルからの読み込み
\fbbox{\rule{0pt}{3zw}\rule{3zw}{0pt}}
```

```
\caption{picture 環境で描画した図形}\label{fig:sample}
\end{center}
\end{figure}
\nocite{*}
\begin{thebibliography}{10}% 参考文献
\bibitem{latexcompanion}
    Michel Goossens, Frank Mittelbach, and Alexander Samarin.
    The \LaTeX コンパニオン.
    東京アスキー, 1998.
\bibitem{latexgraphics}
    Michel Goossens, Sebastian Rahtz, and Frank Mittelbach.
    \LaTeX グラフィックスコンパニオン.
    株式会社アスキー, 2000.
\bibitem{bibunsyo}
    奥村晴彦.
    [改訂第 3 版] {\LaTeXe} 美文書作成入門.
    技術評論社, 2004.
\bibitem{platex2e}
    乙部蔵己, 江口庄英.
    {\em {p\LaTeXe} for Windows Another Manual Vol.1 Basic Kit 1999}.
    ソフトバンク, 1998.
\bibitem{linuxthesis}
    臼田昭司, 伊藤敏, 井上祥史.
    Linux 論文作成術.
    オーム社, 1999.
\bibitem{metafont}
    Donald~E. Knuth.
    \textsf{METAFONT} ブック.
    アスキー, 1994.
\bibitem{jtexbook}
    Donald~E. Knuth.
    改訂新版{\TeX}ブック.
    アスキー出版局, 1992.
\end{thebibliography}
\end{document}
```

2 段組での中間報告のサンプル

システム情報科学部 情報アーキテクチャ学科

m1202147 渡辺 徹

指導教官 未来 太郎

2004 年 9 月 20 日

概要

論文作成においては \LaTeX を使用するのが望ましいが、近年では事務処理用の Word がその代わりとなっているように見受けられる。今回は、はこだて未来大学においてどの程度 Word や \LaTeX が浸透しているのかを 2003 年度の卒業研究から提出される中間レポートを参考に統計を取ってみた。結果は予想通り Word 人口が圧倒的に多かった。また、この中間報告のサンプルの内容は出たら目であるので、あくまで入力例として参考にしてもらいたい。

1 目的

当大学では卒業研究の中間報告として中間レポートを提出するようになっている。各自がどのようなアプリケーションを使っているのかを調査することが今回の目的である。

2 方法

直接研究生にアンケートをとったわけではなく、ウェブページ上で 2003 年 9 月 10 日までに提出されているレポートを調査対象とした。

3 結果

提出されているレポートを大まかに調査した結果が表 1 となる。これは研究生がどのようなアプリケーションで中間レポートを作成したのかを調べた結果である。どうしても判別できないものは**その他**の項目に入れてある。レポートの最終形態ではなく、原稿を作成する段階で使ったアプリケーションを示している。これらの結果は二次的に入手した情報のため、データに若干の誤りがある。直接アンケートをとって調べればもっと正確な情報が収集できるが、今回は簡易的な形をとった。

表 1 データの分析結果

項目	人数 (人)	割合 (%)
Word	75	45.2
\LaTeX	26	15.6
HTML	54	32.5
Illustrator	4	2.4
OpenOffice	1	0.6
その他	6	3.0
合計	166	100

4 考察

以上の結果から、現在 HTML で作成している人物は Word を使う事になるだろう。結果があくまで中間報告である事を考えれば、Word 人口がこれから増えることは明白である。今度の働きかけ次第で当大学の \LaTeX 人口を増加させることも可能である。

この現象を天下り的にフーリエ変換で解析する。まず、フーリエ変換で関数 $f(x)$ を定義する。この関数 $f(x)$ は変換のための区間を必要とするので、区間を $[-L, L]$ とする。すると以下の式が定義から

導出される.

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right)$$

$$a_n = \frac{1}{L} \int_{-L}^L f(u) \cos \frac{n\pi u}{L} du$$

$$b_n = \frac{1}{L} \int_{-L}^L f(u) \sin \frac{n\pi u}{L} du$$

よって, 次式 (1) が新たに得られる.

$$f(x) = \frac{1}{2L} \int_{-L}^L f(u) du$$

$$+ \sum_{n=1}^{\infty} \left[\frac{1}{L} \int_{-L}^L f(u) \cos \frac{n\pi u}{L} du \cdot \cos \frac{n\pi x}{L} \right.$$

$$\left. + \frac{1}{L} \int_{-L}^L f(u) \sin \frac{n\pi u}{L} du \cdot \sin \frac{n\pi x}{L} \right] \quad (1)$$

式 (1) を $L \rightarrow \infty$ にしたりしてフーリエ変換は一般に式 (2) のように書き表すことができる.

$$F(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(u) e^{-i\alpha u} du \quad (2)$$

式 (2) を使って今回の結果を解析することは, 現段階では非常に困難であると容易に考察できる.

5 今後の展望

今回得られた調査結果を下に Gnuplot でデータをプロットする作業が続くものと思われる. また, グラフは主に Gnuplot から挿入するのが望ましいとされる. Gnuplot から挿入したグラフは図 1 となる.



図 1 picture 環境で描画した図形

参考文献

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. The L^AT_EX コンパニオン. 東京アスキー, 1998.
- [2] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. L^AT_EX グラフィックスコンパニオン. 株式会社アスキー, 2000.
- [3] 奥村晴彦. [改訂第 3 版] L^AT_EX 2_ε 美文書作成入門. 技術評論社, 2004.
- [4] 乙部 敏己, 江口 庄英. pL^AT_EX 2_ε for Windows Another Manual Vol.1 Basic Kit 1999. ソフトバンク, 1998.
- [5] 白田昭司, 伊藤敏, 井上祥史. Linux 論文作成術. オーム社, 1999.
- [6] Donald E. Knuth. METAFONT ブック. アスキー, 1994.
- [7] Donald E. Knuth. 改訂新版 T_EX ブック. アスキー出版局, 1992.

10.2 学位論文のサンプル

学位論文などは規模として大きくなるので文書クラスは jreport か jsbook を使うことになります。jsbook の場合にクラスオプションは

```
\documentclass[openany,oneside,11pt]{jsbook}
```

とすると左右起しをせずに片面印刷で出力されます。

jreport や jsbook で使用できる見出しは

章見出し `\chapter`

節見出し `\section`

小節見出し `\subsection`

の三つです。`\subsubsection` 命令はなるべく使わないほうが良いでしょう。jsarticle 文書クラスで使用できた abstract 環境は使えなくなりますので

```
\chapter*{概要}\addcontentsline{toc}{chapter}{概要}
ここに簡潔に概要を書く。
```

として章立てします。

▼ 10.2.1 クラスファイルが提供されている

学位論文などは大学側から文書クラスが提供されることがあります。当大学の卒業論文の場合は funthesis というクラスファイルが配布されていますのでこれを使うことになります。クラスファイルとして funthesis を使った例を示します。出力は省略させていただきます。

```
%\documentclass[english]{funthesis}% 本文が英語のとき
\documentclass{funthesis}
\usepackage[dvipdfm]{graphicx}% dvipsk の場合は‘dvips’にする
% 日本語の題名
% 長いときは‘\’で改行
\jtitle{公立はこだて未来大学における卒業論文の
{\LaTeX}クラスファイルの設計に関する考察}
% 論文の英文タイトル
\etitle{Title in English}
% 氏名(日本語)
\jauthor{未来 太郎}
% 氏名(英語)
\eauthor{Taro MIRAI}
% 所属学科名
\affiliation{複雑系アーキテクチャ学科}
% 学籍番号
\studentnumber{1300000}
```

```

% 正指導教員名
\advisor{正指導 教員}
% 副指導教員がいる場合はコメントアウトし名前を書く
% 副指導教員がいない場合は、ここは削除しても可
%\coadvisor{副指導 教員}
% 論文提出日
\date{2004/01/31}
% ここから本文の始まり
\begin{document}
% 表紙
\maketitle
% 英語の概要
\begin{eabstract} Abstract in English. (about 500 words)
\fake{you should write your English abstract in one page. }
% 英文キーワード
\begin{ekeyword}
Keyrods1, Keyword2, Keyword3, Keyword4, Keyword5
\end{ekeyword}
\end{eabstract}
% 和文概要 (2000 字程度)
\begin{jabstract} 日本語の概要を書く。(約 200 字)
\fake{ここに日本語の概要を書きます。}
% 和文キーワード
\begin{jkeyword}
キーワード 1, キーワード 2, キーワード 3, キーワード 4, キーワード 5
\end{jkeyword}
\end{jabstract}
% 目次
\tableofcontents
% 図目次
\listoffigures
% 表目次
\listoftables
\chapter{序論} % 章 (chapter) のタイトル
ここに序論を書きます。
\section{背景} % 節 (section) のタイトル
以下に背景, 関連する環境, 状況, 技術に関する概要を記述。
\chapter{考察}
考察しました。
\section{評価結果}
評価結果をここに記述します。
\chapter{結論と今後の展開}
結論と今後の展望をここに記述します。
% 以降, 付録 (付属資料)であることを示す
\begin{appendix}
\chapter{アルゴリズム}
% 付録その 1 (関連資料など)を必要があれば載せる
\section{あるアルゴリズム}

```

```
% 付録その2(関連資料など)を必要があれば載せる
\chapter{ソースコード}
プログラムのソースコードなどを掲載します.
\section{あるソースコード}
何かを処理するあるプログラム\texttt{hoge.cpp}のプログラムを示す.
\begin{verbatim}
int main( void ){ return 0; }
\end{verbatim}
\fake[40]{\thehoge} lines \par}
% 付録の終わり
\end{appendix}
\chapter*{謝辞}
謝辞を書く.
% 参考文献
\begin{thebibliography}{9}
\bibitem{ラベル} 著者名. 書籍名. 出版社, 年号.
\bibitem{MT1999} 未来太郎. 未来の未来. どこかの出版, 1999.
\end{thebibliography}
\end{document}
```

▼ 10.2.2 クラスファイルが提供されていない

もし大学側からクラスが提供されていない場合は自前で作成することになります。しかも大抵の大学は Times 系のフォントを使ってフォントサイズは何々でという細かい指定をしてくるのが普通ようです。親切な教員が作成してくれている場合もあります。とりあえず子供だましですが jsbook を用いた例を紹介します。jreport を使っても良いですが jsbook の方が個人的には良いと感じています。まずはご自分の大学の規定に合わせて jsbook に定義のいくつかに変更を加えます。jsbook そのものに変更を加えるとどこにどのような変更を加えたのかが分からなくなる問題などがありますので、別ファイル mygs.sty に変更したマクロなどをまとめておきます。ファイルの先頭に

```
%% File: mygs.sty
%% Copying : Your Name
%% E-mail : name@univ.ac.jp
%% Date : 2004/02/20
\ProvidesPackage{mygs}[2004/03/31 First Family]
```

のようなファイル情報を書き込んでおくといよいでしょう。大抵の機関で Times 系のフォントを指定すると思いますので

```
\RequirePackage{txfonts}
```

の 1 行も必要でしょう。マクロパッケージの中で他のパッケージを必要とする場合は \Requirepackage 命令を使います。

まずは 1 行の字数です．1 行 40 文字であったとすると長さ `\textwidth` に全角 40 文字の幅 (40zw) を指定します．

```
\setlength\textwidth{40zw}
\setlength\fullwidth{\textwidth}%jsbook で必要
```

行数は 40 行と指定されている場合 `\textheight` に 40 行送り分 (40`\baselineskip`) を指定します．

```
\setlength\textheight{40\baselineskip}
```

この程度でも良いと思うのですが，

```
\setlength\hoffset{13\p@}
\setlength\voffset{0\p@}
\setlength\evensidemargin{0\p@}
\setlength\oddsidemargin{\evensidemargin}
\setlength\topmargin{0\p@}
\setlength\headheight{0\p@}
\setlength\headsep{0\p@}
\setlength\marginparwidth{0\p@}
\setlength\marginparpush{0\p@}
\setlength\marginparsep{0\p@}
```

のように設定しても良いでしょう．ここでの `\p@` は単位 ‘pt’ のことです．マクロの中ではこのような命令を使うと良いそうです．ここでは傍注やヘッダーを出力しないと仮定してほとんどの項目に ‘0pt’ を代入しています．

ヘッダーやフッターは割とシンプルなものなので良いと思うので jsbook の場合は

```
\pagestyle{plainfoot}
```

としてフッターの中央にページ番号を出力するようにします．jreport は最初からシンプルな plain というページスタイルになっています．ただし，‘- 13 -’ のようにダッシュも入れるときは

```
\let\@mkboth\@gobbletwo
\let\@oddhead\@empty
\let\@evenhead\@empty
\def\@oddfoot{\normalfont\hfil-- \thepage\ --\hfil}%
\let\@evenfoot\@oddfoot
\setlength\footskip{2\baselineskip}% 必要に応じて
```

とします．ページ番号を太字にするときは `\normalfont` に後に `\bfseries` を追加します．

見出しのフォントの場合は和文はゴシック，欧文は Times Bold としたい場合は jsbook の場合は

```
\renewcommand{\headfont}{\gtfamily\rmfamily\bfseries}
```

のようにしておけば良いでしょう。jsbook は標準では欧文がサンセリフ体になっています。jreport の場合は最初から欧文がボールド体に設定されています。

おまけに目次の深さを決めるカウンタ tocdepth も

```
\setcounter{tocdepth}{2}
```

とすると\subsection まで出力されます。

jreport の場合は見出しの後の字下げが行われないことがありますので

```
\RequirePackage{indentfirst}
```

として indentfirst パッケージを読み込みます。

これらをまとめると自分のマクロパッケージ mygs.sty が出来上がりです。

```
%% File: mygs.sty
%% Copying : Thor Watanabe
%% E-mail : thor@tex.dante.jp
%% Date : 2004/02/20
\ProvidesPackage{mygs}[2004/02/20 First Family]
\RequirePackage{txfonts}% Times 系のフォントを使う
%\RequirePackage{indentfirst}% jreport は必要
\setlength\textwidth{40zw}%1 行 40 文字
\setlength\fullwidth{\textwidth}%jsbook では必要
\setlength\textheight{40\baselineskip}%1 ページ 40 行
\setlength\hoffset{13\p@}%\p@は 0pt のこと
\setlength\voffset{0\p@}
\setlength\evensidemargin{0\p@}
\setlength\oddsidemargin{\evensidemargin}
\setlength\topmargin{0\p@}
\setlength\headheight{0\p@}
\setlength\headsep{0\p@}
\setlength\marginparwidth{0\p@}
\setlength\marginparpush{0\p@}
\setlength\marginparsep{0\p@}
\setlength\footskip{2\baselineskip}% 必要に応じて
\def\ps@foot{% フッターに ‘-- ページ番号 --’ としたいとき
  \let\@mkboth\@gobbletwo
  \let\@oddhead\@empty
  \let\@evenhead\@empty
  \def\@oddfoot{\normalfont\hfil-- \thepage\ --\hfil}%
  \let\@evenfoot\@oddfoot
}
\pagestyle{plainfoot}%jsbook ならば
%\pagestyle{plain}%jreport ならば
\renewcommand{\headfont}{\normalfont\bfseries}
\setcounter{tocdepth}{2}
```

そのような作業が終わったら自分の論文の主となるソースファイルを書き上げます．用紙は A4 で，フォントサイズは 11 pt，左右起しはせずに片面印刷というのが一般的だと思いますから

```
\documentclass[a4j,11pt,openany,oneside]{jsbook}
```

のようにします．そして先程作成した mygs.sty を

```
\usepackage{mygs}
```

として読み込みます．

この程度でも良いのですが，表紙もまた細かい指定をされる場合があります．1 から \maketitle を作っても良いのですが，一刻も早く論文を仕上げなければならないときに，命令を定義しては間に合わないかも知れません．そのようなときは断腸の思いで \titlepage 環境を借用して表紙を作ることもできます．例として \maketitle 命令の変更例を紹介します．

```
\renewcommand{\maketitle}{%
\begin{titlepage}
  \let\footnotesize\small
  \let\footnoterule\relax
  \let\footnote\thanks
  \null\vskip2em% ページ上部の空白
  \begin{center}\thispagestyle{empty}%
    {\LARGE\headfont ここに表題を書きます}\par\vskip1.5em
    {\Large\normalfont 未来太郎}\par\vskip2em
    {\small 未来研究学科 \quad 学籍番号}\par\vskip1em
    {\small 指導教員 \quad 北海太郎}\par\vskip2em
    {提出日 2004/02/30}\par\vskip1em
    {\Large\headfont English Title}\par\vskip1em
    {\large\rmfamily Your Name}\par\vskip1em
  \end{center}%
  \vfill\null
\end{titlepage}}
```

\vskip とは垂直方向に空気を挿入する命令です．

以上は例ですので先方に規定された通りのレイアウトに適宜変更してください．

付録 A

GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter sec-

tion of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To

"Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give

them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not

be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included

in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.



参考文献

- [1] 江口庄英. *Ghostscript Another Manual*. ソフトバンク, 1997
- [2] Michel Goossens and Sebastian Rahtz. *L^AT_EX Web コンパニオン* T_EX と HTML/XML の統合. アスキー, 2001. 鷺谷好輝翻訳
- [3] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX コンパニオン*. アスキー, 1998. 021.49/Go
- [4] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *L^AT_EX グラフィックス コンパニオン*. アスキー, 2000. 鷺谷好輝訳, 007.63/Go
- [5] George Gratzer. *Math into L^AT_EX*. Birkhauser, 2000. 021.49/Gr
- [6] 日本エディタースクール編. 校正記号の使い方 タテ組・ヨコ組・欧文組. 日本エディタースクール, 1999
- [7] 日本エディタースクール編. 文字の組方ルールブック〈ヨコ組編〉. 日本エディタースクール, 2001
- [8] 藤田眞作. *X_YMT_EX: typesetting chemical structural formulas*. 星雲社, 1997. 430.7/Fu
- [9] 藤田眞作. *L^AT_EX 2_ε コマンドブック*. ソフトバンク, 2003
- [10] Cho Jin-Hwan. *DVIPDFMx, an eXtension of DVIPDFM*, 2003.
▶ <http://project.ktug.or.kr/dvipdfmx/>
- [11] 木下是雄. 理科系の作文技術. 中公新書 624. 中央公論社, 1981. 080//Ch//624
- [12] Donald Knuth. 改訂新版 T_EX ブック. アスキー, 1992. 斎藤信男監修, 鷺谷好輝翻訳
- [13] Leslie Lamport. 文書処理システム L^AT_EX 2_ε. ピアソン・エデュケーション, 1999. 阿瀬はる美翻訳, 021.4//La
- [14] 松井正一. 日本語 BibT_EX: jBibT_EX, 1991. jBibT_EX と共に配布される文書
- [15] 小林道正, 小林研. L^AT_EX で数学を L^AT_EX 2_ε + A_MS-L^AT_EX 入門. 朝倉書店, 1997. 410.7//Ko
- [16] 中田英雄, 金城悟編. 大学生のための研究論文のまとめ方 データ収集からプレゼンテーションまで. 文化書房博文社, 1998. 002.7//Na
- [17] Tobias Oetiker. L^AT_EX 2_ε への道 94 分 L^AT_EX 2_ε 入門, 2000. 野村昌孝訳,
▶ <http://www.ring.gr.jp/pub/text/CTAN/info/lshort/japanese/>
- [18] Tobias Oetiker. *The Not So Short Introduction to L^AT_EX 2_ε*, 2003.
▶ <http://www.ring.gr.jp/pub/text/CTAN/info/lshort/english/>
- [19] 小笠原喜康. 大学生のためのレポート・論文術. 講談社, 2002. 080//Ko//1603

- [20] 大隈秀夫. 分かりやすい日本語の書き方. 講談社, 2004. 080//Ko//1644
- [21] 奥村晴彦. [改訂第 3 版] \LaTeX 2 ϵ 美文書作成入門. 技術評論社, 2004. 021.49/Ok
- [22] 乙部徹己, 江口庄英. *p \LaTeX 2 ϵ for Windows Another Manual Vol.2 Extended Kit 1997*. ソフトバンク, 1997. 021.49/Ot
- [23] 乙部徹己, 江口庄英. *p \LaTeX 2 ϵ for Windows Another Manual Vol.1 Basic Kit 1999*. ソフトバンク, 1998. 021.49/Ot
- [24] Oren Patashnik. \BibTeX ing: \BibTeX の使い方, 1991. 松井正一訳, \jBibTeX と共に配布される文書
- [25] Keith Reckdahl. *Using Imported Graphics in \LaTeX 2 ϵ* , 1997.
 - ▶ <http://www.ring.gr.jp/archives/text/CTAN/info/epslatex.pdf>
- [26] 嶋田隆司. \LaTeX 2 ϵ 数式環境 $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX を使いこなす. シイエム・シイ出版部, 2001. 410//Sh
- [27] 臼田昭司, 伊藤敏, 井上祥史. *Linux 論文作成術*. オーム社, 1999. 816.5/Us
- [28] Mark A. Wicks. *Dvipdfm User's Manual*, 1999.
 - ▶ <http://gaspra.kettering.edu/dvipdfm/>
- [29] Thomas Williams and Colin Kelley. *Gnuplot, An Interactive Plotting Program*, 2003.
 - ▶ <http://www.gnuplot.info/>
- [30] Timothy Van Zandt. *PSTricks: PostScript macros for Generic \TeX* , 1993.
 - ▶ <http://www.tug.org/applications/PSTricks/>

索引

数字/記号

\sqcup	33
数式中の	83
文中の	122
\sqcup	28, 70
$\!$	83, 122
$\" (ü)$	28
$\#$	8
$\#$	8, 70
$\$$	8
$\$$	8, 10, 70, 79
コンソールの	ii
$\$$	80
$\%$	8
$\%$	8, 29, 70
$\&$	8
$\&$	8, 70
array 環境の	88
eqnarray*環境の ..	81
tabular 環境の ..	105
$\' (é)$	28
\prime	30
\langle	79
\langle	87
\rangle	79
\rangle	87
\ast	12
\backslash	83, 122
, (読点)	26
-	31
ハイフンとしての ..	31
マイナスとしての ..	31
--	31
---	31
$\backslash (ä)$	28
. (句点)	26
... (中点 3点リーダー) ..	95
... (下付 3点リーダー) ..	95
/	88, 101
区切り記号の	88
分数の	90
\backslash	83, 122
\backslash	83, 122
\lt	8
$\backslash (=)$	28
\gt	8

$\@$	69
$\@$	33
$\backslash [$	80
$[$	87
\backslash	8, 66, 70
$\backslash]$	80
$]$	87
$\backslash \hat{ } (\hat{o})$	28
$\hat{ }$	8, 70, 83
\backslash	8
\backslash	8, 70, 83
$\backslash \grave{ } (\grave{a})$	28
$\grave{ }$	30
$\backslash \{$	8, 87
$\{$	8, 70
$\backslash \}$	8, 87
$\}$	8, 70
$ $	8, 87
$\backslash \tilde{ } (\tilde{n})$	28
\sim	8, 34, 70, 122
10pt	25, 44
11pt	44
12pt	44
2 項演算子	85
2 段組	117, 127
の段間の罫線	118
のときの段間	118
3 点リーダー	95
下付き	95
中点	95

A

$a4j$	44
$a4paper$	44
$a5j$	44
$a5paper$	44
$\AA (\text{\AA})$	28
$\aa (\text{\aa})$	28
abbrv	55
abstract	127
abstract	21, 23
abstract 環境	23
Acrobat Reader	62
$\acute{ } (\acute{a})$	94
\addcontentsline	23
address (BibTeX)	53

\addtocounter	75
\addvspace	123
Adobe Reader	62
$\AE (\text{\AE})$	28
$\ae (\text{\ae})$	28
\afterpage	46
afterpage	46
Alan Jeffery	iii
$\aleph (\aleph)$	95
\Alph	75
\alph	75
$\alpha (\alpha)$	91
alpha	55
$\amalg (\amalg)$	93
American Mathematical	45
Society	45
amount of substance	26
ampere	26
amsbsy	100
amsfonts	95
$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$	45
amsmath ..	45, 82, 90, 99, 100
amssymb	82, 95
$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$	45
amsthm	96
$\angle (\angle)$	95
annotate (BibTeX)	53
\appendixname	118
$\approx (\approx)$	93
\arabic	75
$\arccos (\arccos)$	85
$\arcsin (\arcsin)$	85
$\arctan (\arctan)$	85
$\arg (\arg)$	85
array	45
array 環境	45, 88, 105
$\Arrowvert (\parallel)$	87
$\arrowvert (\parallel)$	87
article	21, 43, 127
article (文献の種類)	53
$\ast (*)$	93
$\asymp (\asymp)$	93
\AtBeginDvi	61
author	19, 20
author (BibTeX)	51, 53
author	20

.aux (拡張子) 16

B

\b (a) 28
 b4j 44
 b4paper 44
 b5j 44
 b5paper 44
 babel 45
 \backslash (\\) 87, 95
 balance 119
 \bar (\bar{a}) 94
 \baselineskip 136
 \baselinestretch 25
 .bb (拡張子) 16
 .bbl (拡張子) 16
 \begin 14
 \beta (β) 91
 \bfseries 41, 136
 .bib (拡張子) 16
 \bibitem 47, 48, 49
 \bibliography 51, 52
 \bibliographystyle 51, 52
 \bibname 118
 BibTeX 47
 address 53
 annotate 53
 author 51, 53
 booktitle 53
 chapter 53
 crossref 53
 edition 53
 editor 53
 howpublished 53
 journal 53
 key 53
 month 53
 note 51, 53
 number 53
 organization 53
 pages 53
 publisher 51, 53
 school 53
 series 53
 title 51, 53
 volume 53
 year 51, 53
 yomi 51, 55
 \Big 87
 \big 87, 88
 \bigcap (\cap) 93
 \bigcirc (\bigcirc) 93
 \bigcup (\bigcup) 93
 \bigl 87
 \bigr 87
 \bigodot (\bigodot) 93

\bigoplus (\bigoplus) 93
 \bigotimes (\bigotimes) 93
 \bigr 88
 \bigskip 123
 \bigsqcup (\bigsqcup) 93
 \bigtriangledown (\bigtriangledown) 93
 \bigtriangleup (\bigtriangleup) 93
 \biguplus (\biguplus) 93
 \bigvee (\bigvee) 93
 \bigwedge (\bigwedge) 93
 .blg (拡張子) 16
 \bm 100
 bm 46, 100
 \bmod 85
 .bmp (拡張子) 16
 \boldmath 99
 \boldsymbol 100
 book (文献の種類) 53
 booklet (文献の種類) 53
 booktitle (BibTeX) 53
 \bordermatrix 90
 \bot (\bot) 95
 \bowtie (\bowtie) 93
 \Box (\Box) 95
 \bracevert (,) 87
 \breve (\breve{a}) 94
 .bst (拡張子) 16
 \bullet (\bullet) 93

C

\c (c) 28
 Calc 108
 calc 45
 Calc2LaTeX 108
 candela 26
 \cap (\cap) 93
 \caption 75, 104
 \catcode 69
 \cdot (\cdot) 93
 \cdots (\cdots) 95
 \centering 38
 centering 38
 center 環境 38
 \chapter .. 21, 22, 46, 75, 133
 chapter (BibTeX) 53
 chapter (カウンタ) 74
 chapter 21
 \chapter* 23
 character 39
 \check (\check{a}) 94
 \chi (χ) 91
 Cho Jin-Hwan iii, 62
 Chris Rowley iii
 \circ (\circ) 93
 \cite 46, 47, 48, 51
 cite 57

\citeform 57
 \citeleft 57
 \citemid 57
 \citepunct 57
 \citeright 57
 class 43
 \cleardoublepage 117
 \clearpage 46, 117
 \cline 90, 106
 .clo (拡張子) 16
 .cls (拡張子) 16
 \clubsuit (\clubsuit) 95
 Colin Kelley 116
 color 45
 \columnsep 118
 \columnseprule 118
 \columnwidth 119
 comment 29
 comment out 29
 comment 環境 29
 config.pdf 61
 config.ps 61
 \cong (\cong) 93
 contents 19, 22
 \contentsname 118
 \coprod (\coprod) 93
 \cos (cos) 85
 \cosh (cosh) 85
 \cot (cot) 85
 \coth (coth) 85
 \cr 90
 CrateBB 16
 \crrc 99
 crossref (BibTeX) 53
 \csc (csc) 85
 \cup (\cup) 93

D

\d (a) 28
 \dag (\dagger) 28
 \dagger (\dagger) 93
 \dashv (\dashv) 93
 \date 19, 20
 date 20
 David Carlisle iii
 dcolumn 46
 \ddag (\ddagger) 28
 \ddagger (\ddagger) 93
 \ddot (\ddot{a}) 94
 \ddots (\ddots) 95
 \deg (deg) 85
 delarray 46
 \Delta (Δ) 92
 \delta (δ) 91
 description 環境 35
 \det (det) 85

\backslash DH (Đ) 28
 \backslash dh (ð) 28
 \backslash Diamond (◇) 95
 \backslash diamond (◊) 93
 \backslash diamondsuit (◊) 95
 \backslash dim (dim) 85
 displaymath 環境 80
 \backslash displaystyle 91
 \backslash div (÷) 93
 \backslash DJ (Đ) 28
 \backslash dj (đ) 28
 document 19
 \backslash documentclass 14, 17
 document 環境 14
 Donald Arseneau 57, 125
 Donald Knuth iii, 3
 \backslash dot (â) 94
 \backslash doteq (≐) 93
 doublespace 25
 \backslash Downarrow (⇓) 87, 94
 \backslash downarrow (↓) 87, 94
 draft 44
 DTP 60
 .dtx (拡張子) 15
 DVI 5, 7, 60
 の再表示 13
 .dvi (拡張子) 16
 dvioout 12
 Dvipdfm 16, 62
 dvipdfm 63, 64, 109
 Dvipdfmx 62
 dvips 63, 64, 108, 110
 dvipsk 61, 63

E

 e-mail 125
 ebb 16, 63
 edition (BIBTEX) 53
 editor (BIBTEX) 53
 electric current 26
 \backslash ell (ℓ) 95
 em-dash 31
 Emacs 5, 6, 125
 \backslash emph 31
 emphasis 27
 \backslash emptyset (∅) 95
 en-dash 31
 Encapsulated PostScript 108
 \backslash end 14
 \backslash enskip 122
 \backslash enspace 122
 enumerate 46
 enumerate 環境 35, 46
 enumi (カウンタ) 74
 enumii (カウンタ) 74
 enumiii (カウンタ) 74

 enumiv (カウンタ) 74
 EPS 63, 108
 .eps (拡張子) 16
 eps2eps 112
 EPS-Conv 112, 114
 \backslash epsilon (ε) 91
 epstopdf 64
 eqnarray* 環境 81
 eqnarray 環境 75, 81
 \backslash equation 75
 equation (カウンタ) 74
 equation 環境 81
 \backslash equiv (≡) 93
 \backslash eta (η) 91
 \backslash example 67
 Excel 107, 108
 Excel2latex 107
 Excel2tabular 107
 \backslash exists (∃) 95
 \backslash exp (exp) 85

F

\backslash fbox 120
 \backslash fboxrule 120
 \backslash fboxsep 120
 .fd (拡張子) 16
 figure (カウンタ) 74
 figure 104
 figure* 環境 119
 \backslash figurename 118
 figure 環境 104
 final 44
 fixill.pl 113
 \backslash flat (b) 95
 fleqn 44, 80
 float 104
 flushing 38
 flushleft 38
 flushleft 環境 38
 flushright 38
 flushright 環境 38
 \backslash fnsymbol 75
 fontenc 28
 \backslash footnote 27
 footnote (カウンタ) 74
 \backslash footnotemark 107
 \backslash footnotesize 40
 \backslash footnotetext 107
 \backslash forall (∀) 95
 \backslash frac 85, 98, 101
 \backslash framebox 120
 Frank Mittelbach .. iii, 95, 96
 \backslash frontmatter 23
 \backslash frown (⌢) 93
 ftnright 46
 FUNNIST i

 funthesis 133

G

\backslash Gamma (Γ) 92
 \backslash gamma (γ) 91
 \backslash gcd (gcd) 85
 \backslash ge (≥) 93
 \backslash gg (≫) 93
 Ghostscript 60
 giga 26
 GNU
 Emacs 125
 Free Documentation
 License 2
 フリー文書利用許諾契
 約書 2
 Gnuplot 116
 graphicx .. 17, 45, 63, 64, 108,
 109
 \backslash grave (à) 94
 \backslash gtfamily 42
 \backslash guillemotleft («) 28
 \backslash guillemotright (») 28
 \backslash guilsinglleft (‹) 28
 \backslash guilsinglright (›) 28

H

\backslash H (ă) 28
 Harald harders 84
 \backslash hat (â) 94
 \backslash hbar (ħ) 95
 headline 19
 \backslash heartsuit (♥) 95
 hhrline 46
 \backslash hline 90, 105, 106
 \backslash hom (hom) 85
 \backslash hookleftarrow (↵) 94
 \backslash hookrightarrow (↶) 94
 howpublished (BIBTEX) .. 53
 \backslash hphantom 100
 \backslash hspace 117
 \backslash hspace* 117, 122
 \backslash hss 99
 HTML 4, 60, 65
 \backslash Huge 40
 \backslash huge 40
 Hyper Link 60
 HyperTeX 62

I

\backslash i (i) 28
 Illustrator 113
 \backslash Im (ℑ) 95
 ImageMagick 112
 \backslash imath (ı) 95
 \backslash in (€) 93
 inbook (文献の種類) 53

`\include` 123
`\includegraphics` ... 64, 109, 110
`\includeonly` 123
inccollection (文献の種類) . 53
`\indent` 24
indentation 19, 23
indentfirst 24, 46, 137
`\inf` (\inf) 85
info 9
`\infty` (∞) 95
Ingo H. de Boer 126
inproceedings (文献の種類) 53
`\input` 123
.ins (拡張子) 15
`\int` (\int) 93
`\iota` (ι) 91
`\item` 35
itemization 35
itemize 環境 35
`\itshape` 41

J

`\j` (\j) 28
jabbrv 55
JabRef 58
jalpha 55
jarticle 17, 21, 44, 46, 127
jBibTeX 47, 50
jbook 44
jclasses 44
`\jmath` (\jmath) 95
Johannes Braams iii
John Collins 126
John Eaton 116
`\Join` (\bowtie) 95
journal (BibTeX) 53
JPEG 63
.jpg (拡張子) 16
jplain 55
jplain.bst 52
jreport 44, 46, 133
jsarticle 44, 127
jsbook 44, 133, 135
jsclasses 44, 45, 122
jspf 44
junsrt 55

K

`\k` (κ) 28
`\kappa` (κ) 91
Karl Berry 61
kelvin 26
`\ker` (\ker) 85
kerning 33

key (BibTeX) 53
kilo 26
kilogram 26
Kpathsearch 61

L

`\L` (\mathbb{L}) 28
`\l` (\mathbb{l}) 28
`\label` 46, 73, 104, 105
Label ‘key’ multiply defined 78
Label(s) may have changed.
Return to get
cross-fercenses
right. 78
`\Lambda` (Λ) 92
`\lambda` (λ) 91
landscape 44
`\langle` (\langle) 87
`\LARGE` 40
`\Large` 40
`\large` 40
LaTeX 4
原稿 7
処理状況 7
での計算 45
とワープロの違い . 24
における単位 25
に関わるファイル . 15
の動かし方 5
のエラー 10
の基本 5
の最低限の規則 1
の作者 iii
の実行 5
の周辺情報 i
の専門書 iv
の中途ファイル 7
の導入 i
の日本語化 4
の入門書 iv
の入力支援 8
の発音 i
のログファイル 7
LaTeX 35
latex 6
latexmk 52, 126
latexsym 95
layout 46
`\lbrace` ($\{$) 87
`\lceil` (\lceil) 87
`\ldots` (...) 95
`\ldots` ii
`\le` (\leq) 93
leading 34
`\leadsto` (\rightsquigarrow) 95

`\left` 86
`\Leftarrow` (\Leftarrow) 94
`\leftarrow` (\leftarrow) 94
`\leftharpoonup` (\leftharpoonup) .. 94
`\leftharpoonup` (\leftharpoonup) 94
`\leftidx` 84
leftidx 84
`\Leftrightarrow` (\Leftrightarrow) ... 94
`\leftrightarrow` (\leftrightarrow) ... 94
length 26
leqno 44
Leslie Lamport iii, 4
less 9
letter space 34
letterpaper 44
letter サイズ 61
`\lfloor` (\lfloor) 87
`\lg` (\lg) 87
`\lhd` (\lhd) 95
ligature 33
`\lim` (\lim) 85
`\liminf` (\liminf) 85
`\limits` 86
`\limsup` (\limsup) 85
line break 32
List of Figures 22
List of Tables 22
`\listfigurename` 118
`\listoffigures` 22
`\listoftables` 22
`\listtablename` 118
`\ll` (\ll) 93
`\lmoustache` (\hookrightarrow) 87
.lof (拡張子) 16
`\log` (\log) 85
.log (拡張子) 16
`\Longleftarrow` (\Longleftarrow) ... 94
`\longleftarrow` (\longleftarrow) ... 94
`\Longleftrightarrow` (\Longleftrightarrow) 94
`\longmapsto` (\longmapsto) 94
`\Longrightarrow` (\Longrightarrow) .. 94
`\longrightarrow` (\longrightarrow) .. 94
longtable 46
.lot (拡張子) 16
ltboxes.dtx 119
`\ltrans` 84
luminous intensity 26

M

Make 52, 126
`\makeatletter` 69
`\makeatother` 69
`\makebox` 119
`\maketitle` 20, 23, 138
manual (文献の種類) 53

\mapsto (\mapsto) 94
 \marginpar 27
 Mark Wicks iii, 62
 mass 26
 masterthesis (文献の種類) .. 53
 \mathbb{b} 83
 \mathbf{b} 82, 99, 100
 \mathcal{b} 82
 Mathematica 113
 \mathfrak{b} 83
 b 2, 82
 b 82
 b 82, 92
 b 82
 \mathstrut 100
 \mathhtt{b} 82
 math 環境 80
 MATLAB 114
 \mathbf{matrix} 90
 matrix 環境 90
 \max (max) 85
 \mbox{b} 119
 \mcfamily 42
 \mdseries 41
 \medskip 123
 mega 26
 MetaPost 16
 meter 26
 \mho (\mho) 95
 Michael Downes iii
 micro 26
 Microsoft Office 3
 \mid (\mid) 93
 milli 26
 \min (min) 85
 minipage 環境 114, 121
 misc (文献の種類) 53
 Misplaced alignment tab character & 10
 Missing \$ inserted 10
 \models (\models) 93
 mole 26
 month (BibTeX) 53
 Morten Alver 58
 \mp (\mp) 93
 \cdot (拡張子) 16
 mpfootnote (カウンタ) .. 74
 μ (μ) 91
 multicol 46
 \multicolumn 90, 105, 106

N

∇ (∇) 95
 nano 26
 \natural (\natural) 95

\nearrow (\nearrow) 94
 \neg (\neg) 95
 \negthinspace 122
 Nelson Beebe iii
 \neq (\neq) 93
 nest 35
 nested sections 21
 \newcommand 67
 $\newcommand*$ 124, 125
 \newcounter 75
 \newenvironment 68
 \newline 32
 \newpage 117
 \newtheorem 95, 96
 N (N) 28
 n (n) 28
 \ni (\ni) 93
 Nizar Batada 58
 \nocite 52
 \noindent 24
 \nolimits 86
 \nonumber 82
 \normalsize 40
 \nosort 57
 \nospace 57
 \not 93, 99
 note (BibTeX) 51, 53
 note 19, 27
 \notin (\notin) 93
 \notitlepage 44
 ν (ν) 91
 number (BibTeX) 53
 \nwarrow (\nwarrow) 94

O

\emptyset (\emptyset) 28
 \circ (\circ) 28
 Octave 116
 \odot (\odot) 93
 OE (OE) 28
 oe (oe) 28
 \oint (\oint) 93
 okumacro 31
 Ω (Ω) 92
 ω (ω) 91
 \ominus (\ominus) 93
 \onecolabstract 127
 \onecolumn 118
 \onecolumn 44
 \oneside 44
 \oalign 99
 \openany 44
 OpenOffice.org 3
 \openright 44
 \oplus (\oplus) 93
 Oren Patashnik iii, 47

organization (BibTeX) .. 53
 \oslash (\oslash) 93
 \otimes (\otimes) 93
 \overbrace 94
 \overfull 11
 \overleftarrow 94
 \overline 94
 \overrightarrow 94

P

\P (\P) 28
 package 43
 page (カウンタ) 74
 \pageref 73
 pages (BibTeX) 53
 \par 24, 32, 34
 \paragraph 21, 22
 paragraph (カウンタ) .. 74
 paragraph 19, 21
 skip 34
 \parallel (\parallel) 93
 \parbox 120
 \parindent 24
 \part 21, 22
 part (カウンタ) 74
 part 21
 ∂ (∂) 95
 PATH 64
 \path 125
 PDF 60, 63
 ブックマーク 62
 \pdfTeX 59
 \pdfwrite 108
 \pdim 61
 Pehong Chen iii
 Perl 126
 \perp (\perp) 93
 \phantom 100
 \phdthesis (文献の種類) .. 53
 Φ (Φ) 92
 ϕ (ϕ) 91
 photoshop 113
 Π (Π) 92
 π (π) 91
 pico 26
 picture 環境 104
 plain 55
 \LaTeX 4
 platex 6
 \pm (\pm) 93
 \pmatrix 90
 \pmatrix 環境 90
 \pmod 85
 PNG 63
 \cdot (拡張子) 16
 \postchaptername 118

`\postpartname` 118
`PostScript` 60
`\postsectionname` 118
`\pounds` (£) 28
`\Pr` (Pr) 85
`\prec` (\prec) 93
`\preceq` (\preceq) 93
`\prechaptername` 118
`\prepartname` 118
`\presectionname` 118
`\prime` (\prime) 95
`proc` 43
`\prod` (\prod) 93
`\propto` (\propto) 93
`ps2pdf14` 64
`\Psi` (Ψ) 92
`\psi` (ψ) 91
`pTeX` 4
`publisher` (BibTeX) .. 51, 53
`punctuation` 19, 26
`pxdvi` 12
`pxfonts` 100

Q

`\qqquad` 83, 122
`\quad` 83, 122
`quotation` 環境 30
`\quotedblbase` („) 28
`\quotesinglbase` (,) 28
`quote` 環境 30

R

`\r` (ρ) 28
`\raggedleft` 38
`\raggedright` 38
`Rainer Schöpf` iii
`\rangle` (\rangle) 87
`\rbrace` ($\}$) 87
`\rceil` (\lceil) 87
`\Re` (\Re) 95
`Red Hat` 61
`\ref` 46, 73, 105
`Reference ‘key’ on page n`
`undefined` 78
`references` 47
`\refname` 118
`\refstepcounter` 75
`\renewcommand` 67
`\renewenvironment` 68
`report` 43
`\rfloor` (\rfloor) 87
`\rgroup` (\rfloor) 87
`\rhd` (\rhd) 95
`\rho` (ρ) 91
`ridt91.eps` 110
`\right` 86, 87

`\Rightarrow` (\Rightarrow) 94
`\rightarrow` (\rightarrow) 94
`\rightharpoondown` (\rightarrow) 94
`\rightharpoonup` (\rightarrow) 94
`\rightleftharpoons` (\rightleftharpoons) 94
`\rmfamily` 41
`\rmoustache` (\smile) 87
`\Roman` 75
`\roman` 75
`\rotatebox` 111
`\rule` 121

S

`\S` (§) 28
`\sb` 84
`school` (BibTeX) 53
`SciLab` 116
`\scriptscriptstyle` 91
`\scriptsize` 40
`\scriptstyle` 91
`\scshape` 41
`\searrow` (\searrow) 94
`Sebastian Rahtz` iii
`\sec` (sec) 85
`secnumdepth` (カウンタ) .. 22
`second` 26
`\section` .. 21, 22, 46, 75, 133
`section` (カウンタ) 74
`section` 21
`sentence space` 34
`serial number` 21
`series` (BibTeX) 53
`\setcounter` 75
`\setminus` (\setminus) 93
`\sffamily` 41
`\sharp` (\sharp) 95
`showkeys` 46
`SI` 26
`\Sigma` (Σ) 92
`\sigma` (σ) 91
`\sim` (\sim) 93
`\simeq` (\simeq) 93
`\sin` (sin) 85
`\sinh` (sinh) 85
`slides` 43
`\slshape` 41
`\small` 40
`\smallskip` 123
`\smash` 101
`\smile` (\smile) 93
`\sp` 84
`space` 57
`\spadesuit` (\spadesuit) 95
`\sqcap` (\sqcap) 93
`\sqcup` (\sqcup) 93
`\sqrt` 85

`\sqsubset` (\sqsubset) 95
`\sqsubseteq` (\sqsubseteq) 93
`\sqsupset` (\sqsupset) 95
`\sqsupseteq` (\sqsupseteq) 93
`\SS` (SS) 28
`\ss` (ss) 28
`\stackrel` 98
`\star` (\star) 93
`\stepcounter` 75
`.sty` (拡張子) 15
`\subparagraph` 21, 22
`subparagraph` (カウンタ) 74
`subparagraph` 21
`\subsection` 21, 22, 133
`subsection` (カウンタ) .. 74
`subsection` 21
`\subset` (\subset) 93
`\subseteq` (\subseteq) 93
`\substack` 99
`\subsubsection` .. 21, 22, 133
`subsubsection` (カウンタ) 74
`subsubsection` 21
`\succ` (\succ) 93
`\succeq` (\succeq) 93
`\sum` (\sum) 93
`\sup` (sup) 85
`\supset` (\supset) 93
`\supseteq` (\supseteq) 93
`\surd` (\surd) 95
`\swarrow` (\swarrow) 94

T

`\t` (τ) 28
`TI` 28
`tabbing` 環境 105
`\table` 75
`table` (カウンタ) 74
`table` 104
`table*環境` 119
`\tablename` 118
`\tableofcontents` 22, 118
`table` 環境 104
`tabular*環境` 45
`tabularx` 46
`tabular` 環境 45, 104, 105
`\tan` (tan) 85
`\tanh` (tanh) 85
`\tau` (τ) 91
`tera` 26
`TEX` 3
`の読み方` 3
`\TeX` 35
`.tex` (拡張子) 16
`\text` 82, 83
`\textasciicircum` 8
`\textasciitilde` 8

- `\textbackslash` 8
`\textbar` 8
`\textbf` 41
`textcomp` 26
`\textgreater` 8
`\textgt` 42
`\textheight` 136
`\textit` 41
`\textless` 8
`\textmc` 42
`\textmd` 41
`\textmu` 26
`\textquotedbl (")` 28
`\textrm` 41
`\textsc` 41
`\textsf` 41
`\textsl` 41
`\textstyle` 91
`\texttt` 41
`\textwidth` 119, 136
`Tgif` 108
`\TH (P)` 28
`\th (p)` 28
`thebibliography` 環境 16, 47, 48, 118
`\theorem` 95, 96
`theorem` 46, 96
`\theorembodyfont` 96
`\theoremheaderfont` 96
`\theoremstyle` 96
`thermodynamic temperature` 26
`\Theta (Θ)` 92
`\theta (θ)` 91
`\thinspace` 122
`Thomas Williams` 116
`\tilde (ã)` 94
`time` 26
`\times (×)` 93
`\tiny` 40
`\title` 19, 20
`title (BibTeX)` 51, 53
`title` 19, 20
`\titlepage` 138
`titlepage` 44
`.toc (拡張子)` 16
`tocdepth (カウンタ)` 22
`Tomas Rokicki` 61, 113
`tools` 45
`\top (T)` 95
`\triangle (Δ)` 95
`\triangleleft (◁)` 93
`\triangleright (▷)` 93
`\ttfamily` 41
`Tutorial` i
`\twocolumn` 118, 119, 127
`twocolumn` 44
`twoside` 44
`txfonts` 100
`typeface` 39

U

`\u (i)` 28
`Unable to open <file>.pdf` 62
`\unboldmath` 99
`\underbrace` 94
`\underline` 94, 121
`underline` 27
`Unix 系 OS` . 5, 6, 7, 9, 12, 13, 60, 64, 108
`の基本ツール` 9
`\unlhd (⊲)` 95
`\unrhd (⊳)` 95
`unsrt` 55
`\Uparrow (↑)` 87, 94
`\uparrow` 87, 94
`\Updownarrow (⬆)` 87, 94
`\updownarrow (⬇)` 87, 94
`\uplus (⊕)` 93
`\Upsilon (Υ)` 92
`\upsilon (υ)` 91
`URL` 125
`\url` 125
`url` 125
`\urlstyle` 125
`\usepackage` 17

V

`\v (ä)` 28
`\value` 75
`\varepsilon (ε)` 92
`varioref` 46
`\varphi (φ)` 92
`\varpi (ϖ)` 92
`\varrho (ρ)` 92
`\varsigma (ς)` 92
`\vartheta (θ)` 92
`\vdash (⊢)` 93
`\vdots (⋮)` 95
`\vec (ā)` 94
`\vee (V)` 93
`\verb` 29
`verbatim` 46
`verbatim` 29
`verbatim 環境` 29, 46
`\Vert (||)` 87
`\vert (|)` 87
`Victor Eijkhout` 29
`\vline` 90, 105, 106
`volume (BibTeX)` 53
`\vphantom` 100
`\vskip` 138
`\vspace` 117
`\vspace*` 117, 123

W

`Warning` 17
`\wedge (Λ)` 93
`white space` 28
`\widehat` 94
`\widetilde` 94
`Windows` 6, 7, 8
`のプレビューア` 12
`ヘルプ` 8
`WinShell` 126
`word` 19
`space` 34
`\wp (ø)` 95
`\wr (ι)` 93
`WYSIWYG` 3

X

`xdvi` 12
`XEmacs` 5
`XHTML` 65
`\Xi (Ξ)` 92
`\xi (ξ)` 91
`xr` 46
`\xspace` 125
`xspace` 46, 125
`XYTeX` 115

Y

`YaTeX` 125
`year (BibTeX)` 51, 53
`yomi (BibTeX)` 51, 55
`You can't use 'macro`
`parameter character`
`#'` 10

Z

`\zeta (ζ)` 91

あ

`空き` 33, 121
`垂直方向の` 123
`水平方向の` 122
`単語間の` 33
`文間の` 33
`文字間の` 33
`アキュート` 28
`アクセント` 28
`大きい` 94
`数式中の` 94
`小さい` 94
`文中の` 28
`アクセント記号` 28
`アクティブ文字` 70
`アットマーク` 50, 69

阿部昌平 108
 アンダーバー 84
 アンド 8, 10
 アンペア 26

い

意思伝達手段 39
 イタリック体 41
 一段組 44
 一括処理 3
 イニシャルコマンド 14
 入れ子 35
 印刷 53
 できるファイル 7
 印刷所 60
 印刷面 44
 インチ 25
 引用 15, 47
 の引用 30
 雑誌名の 31
 書籍名の 31
 単語の 30
 段落の 30
 複数段落の 30
 文の 30
 本の名前の 31
 引用符 15, 31, 34
 欧文の 15
 和文の 15

う

ウェーブ 28
 ウェブブラウザ 60
 受け 27
 ウムラウト 28
 浦壁厚郎 107
 上付き 84
 上付き文字 70

え

英数入力 6
 英文字 70
 江口庄英 4
 エスケープ文字 70
 エラー
 に対するヘルプ 12
 Label ‘key’ multiply
 defined 78
 Label(s) may have
 changed. Return to
 get 78
 cross-ferences
 right. 78
 Misplaced alignment
 tab character & 10
 Missing \$ inserted . 10

Reference ‘key’ on
 page n undefined ..
 78
 Unable to open
 ⟨file⟩.pdf 62
 You can’t use ‘macro
 parameter character
 #’ 10
 エラーメッセージ 7
 円記号 14
 演算子 93
 の否定 99
 2 項 85, 93
 大型 93

お

欧文
 の引用符 15
 大型演算子 93
 大沢英一 56
 大島利雄 60
 大島利夫 iii
 大野修一 116
 尾田晃 116
 奥村晴彦 i, 4, 44, 55, 126
 起こし 27
 オゴネク 28
 乙部蔵己 4
 オプション
 10pt 25, 44
 11pt 44
 12pt 44
 a4j 44
 a4paper 44
 a5j 44
 a5paper 44
 b4j 44
 b4paper 44
 b5j 44
 b5paper 44
 draft 44
 dvipdfm 63, 64, 109
 dvips ... 63, 64, 108, 110
 final 44
 fleqn 44, 80
 landscape 44
 leqno 44
 letterpaper 44
 nosort 57
 nospace 57
 notitlepage 44
 onecolumn 44
 oneside 44
 openany 44
 openright 44
 space 57

T1 28
 titlepage 44
 twocolumn 44
 twoside 44
 親カウンタ 96
 親ディレクトリ 9

か

会議 53
 改行 24, 28, 32
 array 環境での 89
 eqnarray 環境での 81
 L^AT_EX での 28
 行揃えにおける 38
 行末文字としての 70
 連続した 28
 会議録中の論文 53
 書いたまま出力する 29
 改段落 32
 改丁 117
 回転 111
 図の 109
 文字列の 111
 概要 23
 会話文 15
 カウンタ 73
 の新設 74
 の設定 74
 chapter 74
 enumi 74
 enumii 74
 enumiii 74
 enumiv 74
 equation 74
 figure 74
 footnote 74
 mpfootnote 74
 page 74
 paragraph 74
 part 74
 secnumdepth 22
 section 74
 subparagraph 74
 subsection 74
 subsubsection 74
 table 74
 tocdepth 22
 親 96
 化学
 構造式 115
 式 115
 かぎ括弧 15, 27, 30, 31
 学位論文 19
 学会誌 56
 角括弧 14, 15
 で囲まれた引数 14

- 拡大
 区切り記号の 87
 図の 110
 拡張子 15, 43
 .aux 16
 .bb 16
 .bbl 16
 .bib 16
 .blg 16
 .bmp 16
 .bst 16
 .clo 16
 .cls 16
 .dtx 15
 .dvi 16
 .eps 16
 .fd 16
 .ins 15
 .jpg 16
 .lof 16
 .log 16
 .lot 16
 .mp 16
 .png 16
 .sty 15
 .tex 16
 .toc 16
 確認作業 6
 箇条書き 35
 記号付きの 35
 説明付きの 35
 番号付きの 35
 数値の代入 25
 下線 27, 121
 アクセント 28
 の意味 27
 画像
 ベクトル 16
 可逆圧縮 16
 フルカラー 16
 無圧縮 16
 括弧 15, 86
 のある行列 46
 の足りない状態 12
 合字 33
 AE の 28
 OE の 28
 カテゴリーコード 69
 下点アクセント 28
 かな漢字変換プログラム 6
 カレントディレクトリ 9
 巻 55
 環境 14, 66
 abstract 23
 array 45, 88, 105
 center 38
 comment 29
 description 35
 displaymath 80
 document 14
 enumerate 35, 46
 eqnarray 75, 81
 eqnarray* 81
 equation 81
 figure 104
 figure* 119
 flushleft 38
 flushright 38
 itemize 35
 math 80
 matrix 90
 minipage 114, 121
 picture 104
 pmatrix 90
 quotation 30
 quote 30
 tabbing 105
 table 104
 table* 119
 tabular 45, 104, 105
 tabular* 45
 thebibliography 16, 47, 48, 118
 verbatim 29, 46
 定理型の 96
 問題型の 96
 例題型の 96
 関係子 93
 間違いの修正 42
 カンデラ 26
 巻末 47
 き
 キートップ ii
 キーボード ii
 からの入力 ii
 ギガ 26
 記号 65
 の意味 2
 の重ね合わせ 99
 の使い方 2
 の積み重ね 98
 の分類 65
 アクセント 28
 大きさが可変の 85
 区切り 86
 校正 42
 根号 100
 数学 79, 91
 積分 85
 添え字における 86
 特殊な 28, 66
 偏微分 98
 ルート 100
 記事の検索 21
 机上出版 60
 奇数起こし 117
 木下是雄 48
 疑問符 11
 脚注 27, 40, 125
 2 段組での 46
 minipage 環境での 121
 表中の 107
 行頭
 の字下げ 24
 行送り 25, 136
 行間 34
 行揃え 38
 強調 3, 27, 40
 の中の単位 26
 見出しの 42
 文字列の 27
 和文の 27, 42
 行末 122
 のコンマ 51
 文字 70
 行列
 中の小数点 46
 に付ける括弧 46
 の罫線 46
 局所変数 71
 ギリシャ文字 91, 100
 の大文字 91
 の小文字 91
 の変体小文字 92
 キロ 26
 キログラム 26
 禁則処理 42
 く
 空行 28, 79
 空白 70
 行間の 33
 四分空き 34
 段落間の 33
 区切り 86
 記号 86
 項目の 57
 単語の 31
 文の 30
 ページの 117
 くさび 28
 句読点 19, 26
 国 68
 組版 1, 2, 5
 クラス 16, 43, 44
 ファイル 16
 article 21, 43, 127

funthesis 133
 jarticle 17, 21, 44, 46, 127
 jbook 44
 jclasses 44
 jreport 44, 46, 133
 jsarticle 44, 127
 jsbook 44, 133, 135
 jsclasses 44, 45, 122
 jspf 44
 proc 43
 report 43
 slides 43
 欧文専用の 43
 学会誌用の 44
 小規模な文書用の 44
 書籍用の 44
 標準的な 43
 報告書用の 44
 クラスオプション 44
 グループ 66, 79, 80
 グループ
 の終わり 70
 の開始 70
 グレーブ 28
 け
 警告 11, 17
 Overfull 11
 煩雑な 11
 罫線 121
 の太さ 120
 行列の 88
 表中の 105
 ケルビン 26
 研究 1
 原稿 5
 作成時の注意 8
 作成の支援 125
 中の空白 28
 の校正 42
 の入力支援 8
 の汎用性 i
 の分割 123
 の編集 5
 原点座標 63
 こ
 鉤形符 28
 考察 1
 校正 42
 合線 28
 光度 26
 公立はこだて未来大学 i
 コード
 カテゴリー 69
 分類 69
 国際単位 26

黒板太字書体 82
 ゴシック体 42
 コマンド 5, 8, 14
 の後の空白文字 46
 の情報表示 8
 の引数 14
 のヘルプの表示 8
 ラインオプション .. 9, 61
 イニシャル 14
 宣言型の 72
 内部 8
 プリアンブル 14
 命令型の 72
 コマンドプロンプト ii, 6, 8
 コメント 29
 アウト 29
 文字 70
 コロン 27
 コンソール ii
 コントロール
 シークエンス 66
 シンボル 66
 スペース 66
 ワード 66
 コンパイル 7
 コンパニオンシリーズ 4
 コンピュータ i
 コンマ 17, 27
 全角の 26
 さ
 サーカムフレックス 28
 サイズ 41
 再表示機能 13
 索引
 の見出し 118
 作者 20
 作文 2
 技術 48
 雑誌
 和文の 31
 雑誌名
 の引用 31
 論文誌 53, 55
 の巻 55
 の番号 55
 左右起し 44
 参考文献 47, 118
 データベース 47
 の補足事項 49
 の見出し 118
 サンセリフ体 41
 し
 シェイプ 41
 イタリック 41

スモールキャピタル 41
 スラント 41
 シェル ii, 6, 8
 字間 33
 時間 26
 字下げ 19, 23
 の抑制 24
 行頭の 24
 段落始めの 23
 箱の中での 120
 見出し直後の 46
 下付き 84
 文字 70
 字詰め 33
 質量 26
 四分空き 42
 紙面
 構成 1
 の拡大縮小 25
 シャープ S 28
 斜体 3
 住所 68
 修飾子 26
 修士論文 53
 出版社 49, 53, 55, 60
 出版年 55
 章 19, 118
 の見出し 118
 商業出版 3
 小数点 46
 章立て 23, 103, 133
 小なり 8
 章標 28
 情報 19
 情報システム i
 ショート 28
 書式 16, 43
 原稿の 16
 書籍 1
 和文の 31
 書籍名
 の引用 31
 書体 ii, 39
 の種類 ii
 の属性 16
 書名 15, 49, 55
 処理
 の中断 11
 シリーズ 41, 49, 55
 ボード 41
 メディアム 41
 資料の配布 60
 シングルクオート 15, 30
 人名
 Alan Jeffery iii
 Cho Jin-Hwan iii, 62

Chris Rowley iii
 Colin Kelley 116
 David Carlisle iii
 Donald Arseneau 57, 125
 Donald Knuth iii, 3
 Frank Mittelbach . iii, 95,
 96
 Harald harders 84
 Ingo H. de Boer 126
 Johannes Braams iii
 John Collins 126
 John Eaton 116
 Karl Berry 61
 Leslie Lamport iii, 4
 Mark Wicks iii, 62
 Michael Downes iii
 Morten Alver 58
 Nelson Beebe iii
 Nizar Batada 58
 Oren Patashnik ... iii, 47
 Pehong Chen iii
 Rainer Schöpf iii
 Sebastian Rahtz iii
 Thomas Williams ... 116
 Tomas Rokicki .. 61, 113
 Victor Eijkhout 29
 阿部昌平 108
 浦壁厚郎 107
 江口庄英 4
 大沢英一 56
 大島利雄 60
 大島利夫 iii
 大野修一 116
 尾田晃 116
 奥村晴彦 . i, 4, 44, 55, 126
 乙部徹己 4
 竹野茂治 116
 中尾誠 107
 中野賢 iii, 4
 平田俊作 iii
 広瀬雄二 125
 藤田眞作 4, 115
 山賀正人 116
 渡辺徹 iv

す

図 118
 の中央揃え 103
 の見出し 118
 目次 118
 数学関数 84
 数学記号 79, 91
 数式
 中の空白の調節 83
 中の太字 46
 の位置 44

 の組版 79
 の書体の変更 82
 の中の文章 83
 の左揃え 80
 の表示形式の調整 . 90
 の太字 99
 番号の位置 44
 モード 79
 番号付きの 81
 複数行の番号付き .. 81
 文中 79
 別行 80
 数式モード 79
 スクリーンキャプチャ 63
 スコープ 71
 変数の 71
 スタイル
 参考文献 16
 文献一覧の 47
 スタイルファイル 15
 図表
 の配置 104
 見出し 103
 図見出し 103
 スモールキャピタル体 41
 図目次 22, 118
 スラッシュ付き O 28
 スラント体 41

せ

制御綴り 66
 成形 5
 ファイル 7
 整形 5
 製本 53
 積分記号 85
 セクション 28
 節 19
 絶対的な長さ 25
 接頭語 26
 セディラ 28
 セミコロロン 27
 全角ダーシ 31
 全角入力 6
 宣言 66
 型のコマンド 72
 センチメートル 25

そ

相互参照 73
 できるもの 73
 に関わる警告 78
 に必要なファイル ... 7
 の工夫 75
 の仕組み 73
 のラベルの表示 46
 別の文書との 46

装置 45
 添え字 83
 上付きの 83
 下付きの 83
 ソース 16
 ソースファイル ... 5, 7, 16, 43
 の編集 5

た

ダーシ
 全角 31
 波 31
 倍角 31
 ターミナル 6, 7, 8
 タイ 28
 第 118
 大域変数 71
 大学 53
 大なり 8
 タイプセット 5
 後の生成ファイル ... 7
 の終了 12
 の続行 12
 ノンストップの 12
 タイプライタ体 41
 題名 20, 55
 ダガー 28
 高さ
 x の字の 25
 図の 109
 ルートの 100
 竹野茂治 116
 多段組 46, 118
 ダッシュ 31
 の両隣 32
 縦棒 8
 ダブルクオート 27, 30
 ダブルクリック 6
 ダブルスペース 25
 ダブルダガー 28
 単位 25, 26
 のずれ 25
 の接頭語 26
 段組 44
 短音符 28
 短剣符 28
 単語 19
 間の空白 33
 の引用 30
 段標 28
 段落 19
 間の空白 33
 の引用 30

ち

地域 68
 中央揃え 38

注記 49
 注釈 19, 27
 の位置 27
 の役割 27
 中途ファイル 7
 長音符 28
 著作権法 47
 著作物 47
 著者名 55
 の頭文字 51
 の姓名のあいだ 51
 の統一 49
 の読み 55
 チルダ 8, 28

て

ディエレシス 28
 定数 91, 92
 定理 46
 ディレクトリ 8
 の移動 8
 の新規作成 8
 の作り方 8
 親 9
 カレント 9
 テキストエディッタ ii, 5, 6
 Emacs 6
 メモ帳 6
 テキストファイル 5
 テキストモード 79
 デバイス 45
 ドライバ 17
 手引書 i
 テラ 26
 点 95
 のない i 28
 のない j 28
 句読 26
 転載 47
 電子文書形式 60
 電流 26

と

通し番号 21
 ドキュメントクラス 43
 オプション 16, 17
 独自性 1
 読者 47
 特殊記号 29
 特殊文字 28
 ドラフト 44
 ドル 8, 10

な

中尾誠 107
 長さ 26
 の単位 25

1 列の 89
 絶対的な 25
 中野賢 iii, 4
 ナノ 26
 波音符 28
 波括弧 8, 14, 15, 50, 71
 で囲まれた引数 14
 波ダーク 31
 並び替え 50
 引用順の 55
 文献の 50
 難解な用語 19
 ナンバー 8, 10

に

二重引用符 15
 二重かぎ括弧 15, 31
 二重括弧 31
 二重短剣符 28
 二段組 43, 44
 日本語 6
 入力 6
 の幅 25
 日本語化 6
 入力 6
 英数 6
 全角 6
 日本語 6
 半角 6
 入力支援 9
 入力通りの文字の出力 29
 任意引数 14

ね

熱力学温度 26

の

ノンストップ 12

は

場合分け 89
 バージョン情報 7
 パーセント 8, 29
 パーレン 27
 倍角ダーク 31
 媒体 39
 ハイパーリンク 60
 ハイフネーション 31
 ハイフン 31
 の両隣 32

配列

 の要素の区切り 70
 バウンディングボックス 62, 63, 113
 博士論文 53
 箱 119
 広範囲な 120

ハチェック 28
 バックグラウンド 6, 13
 バックスラッシュ 8, 10, 14
 パッケージ 15, 43
 オプション 17
 abstract 127
 afterpage 46
 ambsy 100
 amsmath 45, 82, 90, 99, 100
 amssymb 82, 95
 amsthm 96
 array 45
 babel 45
 balance 119
 bm 46, 100
 calc 45
 cite 57
 color 45
 comment 29
 dcolumn 46
 delarray 46
 doublespace 25
 enumerate 46
 fontenc 28
 ftnright 46
 graphicx 17, 45, 63, 64, 108, 109
 hline 46
 indentfirst 24, 46, 137
 latexsym 95
 layout 46
 leftidx 84
 longtable 46
 multicol 46
 okumacro 31
 pxfonts 100
 showkeys 46
 tabularx 46
 textcomp 26
 theorem 46, 96
 tools 45
 txfonts 100
 url 125
 varioref 46
 verbatim 46
 xr 46
 xspace 46, 125
 発行月 53
 発行年 49, 53
 バッチ処理 3
 バッチモード 12
 ハット 8, 84
 発表年 55
 幅

M の字の 25
 画像の 110
 行列の 89
 要素の 101
 パラグラフ 28
 パラメータ 25
 文字 70
 版 5, 55
 半角記号 8
 半角入力 6
 番号
 箇条書きの 35
 脚注の 107
 論文誌の 53
 参考文献の 49
 数式の 81
 図の通し 104
 図表の通し 104
 図表見出しの通し .. 74
 通し 21
 表の通し 104
 複数行の数式の 81
 見出しの通し 74
 番号付き数式 81
 ひ
 引数 9, 14
 角括弧で挟まれた .. 14
 コマンドの 14
 波括弧で挟まれた .. 14
 ピコ 26
 非常に長い表 46
 左揃え 38
 数式の 80
 日付 20
 必須引数 14
 ビットマップ画像
 可逆圧縮 16
 フルカラー 16
 無圧縮 16
 否定
 演算子の 93
 表 118
 中の小数点 46
 の罫線 46
 の中央揃え 103
 の見出し 118
 目次 118
 長い 46
 ページを跨ぐ 46
 秒 26
 表組み 103
 欧文の 103
 表罫線 103
 表紙の作成 138
 表題 19, 20, 44, 53

同ページの 44
 独立ページの 44
 表紙の 138
 表見出し 103
 表目次 22, 118
 平田俊作 iii
 ビリオド 27
 全角の 26
 広瀬雄二 125
 ふ
 部 19, 118
 の見出し 118
 ファイル
 に付ける名前 112
 の情報表示 8
 の名前の変更 8
 config.pdf 61
 config.ps 61
 DVI 7
 EPS 109
 jplain.bst 52
 ltboxes.dtx 119
 ridt91.eps 110
 印刷できる 7
 画像 104
 組版後の 7
 ソース 5
 ドキュメントクラス 16
 文献スタイル 55
 文献データベース .. 50
 文書クラス 16, 80
 ログ 7
 ファミリー 41
 サンセリフ 41
 タイプライタ 41
 ローマン 41
 フォルダ 8
 の移動 8
 の新規作成 8
 の作り方 8
 フォント
 のアウトライン化 108
 の大きさ 41
 Type1 形式の 61
 ギザギザの 113
 ビットマップ 113
 付加情報 53
 複数行数式 81
 複数行番号付き数式 81
 複数段落の引用 30
 藤田眞作 4, 115
 伏字 L 28
 物理量 26
 符点 28
 浮動体 104

の位置指定 104
 太字 99
 のページ番号 136
 数式中の 46
 ブラックボードボールド体 82
 プリアンブル 17, 51
 コマンド 14
 フリーソフトウェア財団 2
 プレビュー 3, 6
 Red Hat での 12
 Unix 系 OS での ... 12
 Windows での 12
 プレビューア 12
 付録 118
 の見出し 118
 プログラミング言語 3
 プログラム
 Acrobat Reader 62
 Adobe Reader 62
 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ 45
 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 45
 $\mathcal{B}\mathcal{I}\mathcal{B}\mathcal{T}\mathcal{E}\mathcal{X}$ 47
 Calc 108
 Calc2LaTeX 108
 CrateBB 16
 dviout 12
 Dvipdfm 16, 62
 Dvipdfmx 62
 dvipsk 61, 63
 ebb 16, 63
 Emacs 125
 eps2eps 112
 EPS-Conv 112, 114
 epstopdf 64
 Excel 107, 108
 Excel2latex 107
 Exel2tabular 107
 fixill.pl 113
 Ghostscript 60
 Gnuplot 116
 HyperTeX 62
 Illustrator 113
 ImageMagick 112
 info 9
 JabRef 58
 $\mathcal{J}\mathcal{B}\mathcal{I}\mathcal{B}\mathcal{T}\mathcal{E}\mathcal{X}$ 47, 50
 $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ 4
 latex 6
 latexmk 52, 126
 less 9
 Make 52, 126
 Mathematica 113
 MATLAB 114
 MetaPost 16
 Microsoft Office 3
 Octave 116

- OpenOffice.org 3
pdfL^AT_EX 59
pdvips 61
Perl 126
photoshop 113
latex 6
PostScript 60
ps2pdf14 64
pxdvi 12
SciLab 116
T_EX 3
Tgif 108
WinShell 126
xdvi 12
X_YL^AT_EX 115
YaTeX 125
プログラムリスト 29
文 19
の引用 30
分音符 28
文間空白 33
文献 33
の探し方 i
文献一覧 58
の管理 58
の簡略表記 55
文献スタイル 55
abbrv 55
alpha 55
jabbrv 55
jalpha 55
jplain 55
junsrt 55
plain 55
unsrt 55
文献データベース 50
文献の種類 53
article 53
book 53
booklet 53
inbook 53
incollection 53
inproceedings 53
manual 53
masterthesis 53
misc 53
phdthesis 53
文書 19
整形 1
の概略 21
のページレイアウト 46
組版後の 6
ビジネス 38
文章 2
表現 2
文書クラス 43
オプション 17
分数 85
の書き方 90
連 90
文中数式 79
へ 45
米国数学会 45
ページ 16, 59
記述言語 117
の先頭での空き 117
の末尾での空き 120
のような箱 117
改 44
表題 117
ページ区切り 46
ページレイアウト 108
ベクトル 93
画像 16
ベクトル画像 29
べた書き 79
別行数式 ヘルプ
エラーに対する 12
簡易の 9
詳細な 9
編集 5
原稿の 5
ソースファイルの 53
編集者 25, 71, 92
変数 71
の有効範囲 91
変体文字 98
偏微分記号 25
ポイント 85
法 27
傍注 41
ボールド体 19
補足情報の追加 15
補足説明 28
ポント 31
本の名前の引用 ま
マークアップ 1, 39, 40, 43, 60
マイクロ 26
マイナス 31
まえがき 23
前書き部分 17
マクロ 15, 43
の再定義 67
の作成 98
の定義 67
パッケージ 45
数学系の 15
パッケージ化された 15
マクロパッケージ 53
マニュアル 60
配布 15, 27
丸括弧 み
右揃え 38
見出し 19, 21
の作成 21
の直後 46
の通し番号 22
の深さ 21
の変更 118
目次用の 21
ミディアム体 41
ミリ 26
ミリメートル 25
明朝体 42
む 70
無効文字 70
無視される文字 め
命令 14, 66
型のコマンド 72
メートル 26
メガ 26
メモ帳 5, 6
も 79
モード 79
数式 79
テキスト 79
目次 19, 22, 118
の作成 73
の番号付けの深さ 22
の深さ 22
の見出し 118
の見出しの変更 118
用の中途ファイル 16
用の見出し 21
secnumdepth 22
tocdepth 22
図 22, 118
表 22, 118
文字 2, 19, 39
間空白 33
サイズ 44
モル 26
や 94
矢印 116
山賀正人 15
山括弧

よ

揚音符	28
用語	19
の統一	124
用紙	
の大きさ	44
の大きさの指定	61, 62
のサイズ	44
の方向	44
B 列の	61
欧文標準の	61
抑揚音符	28
横罫線	90, 106
読み	55
読み方	3
予約文字	66

ら

ラベル	73
-----	----

り

利用許諾契約書	2
リング	28

れ

列指定子	89, 105
連分数	90

ろ

ローマン体	41
ログファイル	7, 16
ロング	28
ハンガリアン	28

論文

作成	48
における図表	106
会議録中の	53
科学技術系の	i
修士	53
博士	53

わ

ワープロソフト	3
枠	
と文字の間隔	120
の太さ	120
渡辺徹	iv
和文	
の引用符	15

L^AT_EX による論文作成の手引き 2004 年度版

© 渡辺徹 2003, 2004

発行日	2003 年 11 月 第 1 版発行 2004 年 11 月 第 2 版発行 2005 年 3 月 20 日 誤植訂正版配布
-----	---

編集	FUNNI ST
監修	<small>わたなべとある</small> 渡辺 徹 (thor@tex.dante.jp)
装丁	<small>きむらけんいち</small> 木村健一
サポートページ	http://tex.dante.jp/
