

好き好き L^AT_ΕX 2_ε 初級編

渡辺徹
第 1.14 版
2016 年 7 月 23 日

Love Love L^AT_ΕX

— for all beginners at the entry level —

by Thor Watanabe

“The T_EXbook has good examples, problems and jokes.”

Thór Watanabe
<http://github.com/thortex/>
<http://mytexpert.osdn.jp/>

Copyright © 2004, 2005, 2006, 2016 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 1.2 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します. 変更不可部分, 表カバーテキスト, 裏カバーテキストは指定しません. この利用許諾契約書の複製物は GNU *Free Documentation License* (GNU フリー文書利用許諾契約書) という章 (付録 D) に含まれています.

本書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

これは何のための本か

何らかの文書を執筆するときに、まず何を書くべきかという内容に関して考えると思います。しかし、書くべき内容が決まったとしてもどのように書くかは必ずしも決まりきったものではないと思います。大変大きな文書であったり、数式を大量に含むようなものであれば、何かしらの包括的な方法があれば便利でしょう。特に科学技術系のレポートや論文を執筆する事を考えると、それに特化した手法を用いた方が利便性も向上するでしょう。このような状況で広く使われているのが \LaTeX と呼ばれるプログラムです。

この本では \LaTeX を用いた文書作成について解説します。 \LaTeX は \TeX と呼ばれるプログラムの上に構築されているシステムです。 $\text{\TeX}/\text{\LaTeX}$ はフリーウェアであり、誰でも無料で自由 (*free*) に入手する事が可能です。 \LaTeX をうまく使いこなせば、体裁の整った美しい文書が簡単に作成できるようになります。

科学技術系の文書を執筆している時、本来ならば「見出しはゴシック体で 24 pt」であるとか「1 行の文字数は 40 文字で 1 ページは 36 行」という様な書式に関する問題は考えない方が良いでしょう。また、 \LaTeX ではこれが可能です。 \LaTeX では文書の論理構造に気をつけながら原稿を執筆する事ができるため、書式と内容を分離する事が可能なのです^{*1}。

本書は全く \LaTeX を使った事がない人を対象に、すでに必要とされる書式が整っている段階^{*2}で、レポートや論文等の文書を \LaTeX を用いてどのように執筆するかを解説するのが目的となります^{*3}。

この本は単に「 \LaTeX というプログラムの機能を紹介した説明書」というよりは、「動作原理や仕組み理解して \LaTeX を使いこなすための教科書」に近いと思います。昨今

*1 HTML & CSS や XML & XSL のような関係と似ていると思って良いでしょう。

*2 比較的規模の大きな学会等であれば、論文投稿における \LaTeX の書式を提供している所が数多くあります。大学等の教育機関でも \LaTeX 用の学位論文のスタイルを提供している所もあります。

*3 体裁を調整するというのは、本来ならば執筆者が担当する部分ではなく、投稿を求める方が行うべき作業ですから、体裁調整に関する説明は応急処置的な領域に留めています。

は tips 集のような説明書が重宝されがちですが、このような説明書では自力で問題を解決する能力の獲得や、更なる飛躍へ向けたステップアップが難しいという側面があります。TEX や L^AT_EX は何も数式の再現性が素晴らしいからとか、文書に明確な論理構造が求められる状況に効率よく対応するためだけに使われるシステムではないと思います。本書よりも高度な内容に関しては、著者がウェブページで公開している『好き好き L^AT_EX 2_ε』シリーズを参照してください*4。2006 年 4 月現在多くの続編は未完ですが、草稿段階の文書を暫定的に公開しています。

凡例

本書では書体を変更する事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。‘dvi_{pd}fm_x’ という語があったとしても ‘dvi_{pd}fm_x’ や ‘dvi_{pd}fm_x’, ‘dvi_{pd}fm_x’, ‘dvi_{pd}fm_x’ 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 3.20 節を参照してください。

書体	意味	例
ローマン体	通常の文章	dvi _{pd} fm _x
サンセリフ体	パッケージやクラス (3.22 節参照)	dvi _{pd} fm _x
タイプライタ体	キーボードからの入力など	dvi _{pd} fm _x
イタリック体	変数や強調	dvi _{pd} fm _x
スラント体	オプション (3.22.2 節参照)	dvi _{pd} fm _x

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。

本文中の入出力例に対しては、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事をお勧めします。

*4 <http://tex.dante.jp/typo/>

なぜ実行結果を吟味する事を勧めるのか、それはコンピュータプログラムというのは他の分野に比べると現象の再現性が確定的であり、追試可能性が高いという点にあります*5。

科学的な論述が中心の場合は「論拠がなければ信用できない」というのが筋ですが、コンピュータプログラムの場合は、実際に手を使ってプログラムを組んでみて、コンパイルし、実行（実験）し、結果を吟味（考察）する事が誰でも自由にできます（そのプログラムのソースコードが呈示されている限り）。さらに数学の証明のように、アルゴリズム的な話は一度自分でその構造と原理を考えてプログラムすれば、得られた知識は一生その人のものになるのです。このような考えに基づき、本書では例を中心に話を進めているので、その例を自分で入力し、さらにはその実行結果を吟味する事を推奨します。

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 input.tex
$ jbibtex input.tex
$ dvipdfmx -S -o output.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事を示すには `Alt` のようにしています。

`Ctrl+Alt+Delete` は `Ctrl`, `Alt`, `Delete` キーを同時に押す事になります。
`Ctrl+x` `Ctrl+s` は `Ctrl+x` を押した後に `Ctrl+s` を押す事を表します。

本書に表記されているバックスラッシュ '\ ' は Windows 環境によっては、テキストエディタ等で円記号 '¥' として視認できると思います。Windows ユーザの方は基本的に '¥' が Unix 系 OS では '\ ' に文字化けしていると認識して頂いて問題ありません。

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。



少々難解だと思われる箇所、 \LaTeX の動作原理に触れている段落に関しては、



この段落のように『急カーブあり危険!』の \LaTeX マークを付与しています。

*5 心理学実験のように被験者を集める必要性もありません。恐らくある程度の処理能力を有する計算機が一台あれば十分です。

フリーウェアとは

L^AT_EX はフリーウェアですがその重要なマニュアルはフリーではありません。L^AT_EX プロジェクトメンバーの Michel Goossens 氏や Sebastian Rahtz 氏, Frank Mittelbach 氏, Leslie Lamport 氏らが出版しているマニュアルは日本語訳で 1 冊 5,000 円程度の値段です。そこで L^AT_EX ユーザが必携といわれている書籍は 4 冊程あります。

- 『文書処理システム L^AT_EX 2_ε』 [33] 3,000 円。
- 『L^AT_EX コンパニオン』 [34] 4,800 円。
- 『L^AT_EX グラフィックスコンパニオン』 [35] 5,400 円。
- 『L^AT_EX Web コンパニオン』 [36] 4,800 円。

「必携の本を買ったら 18,000 円もかかるのか」と思われる事でしょう。これではフリーウェアだから使ってみようと思った方や、誰かに薦められて使い始めた方は手を出しづらいのではないかと思います。また、L^AT_EX の使い方である技術資料は全て公開されていますので、親切な方がウェブページなどで詳しく取り扱っている場合があります。そのようなページを見れば特に困る事はないと思いますが、情報が離れ離れで存在するので、どうも勝手が悪いようです。

Richard Stallman 氏の訴える通り、これがフリーウェアの抱える問題点ではないかと思えます。そこで新たにフリーなマニュアルを作成する事にしました。ただし L^AT_EX の既存のマクロ、クラスならびにプログラムの活用方法についての話に限定します。事務的な書類の作成ではなく主にレポートや論文を書くための情報を集めていますので、表に色を付けたいとか、フォントにこだわりたいという情報は含んでいません。さらにマクロ・クラスの作成方法は最小限にとどめていますので、既存の良書を付録 B から参照してください。

謝辞

本書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が難しかった事を容易に想像できます。

まず $\text{T}_{\text{E}}\text{X}$ の作者である Donald Knuth 氏には最大の感謝を表さなければなりません。氏が $\text{T}_{\text{E}}\text{X}$ という土台を作ってくれたお陰で，こんなにも素晴らしい世界を体験する事ができた事に喜びを感じております。

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 全般に関しては秋田純一氏，奥村晴彦氏，野村昌孝氏，吉永徹美氏より多くの事を学びました。出版，校正，デザインなどに関しては木村健一氏よりご助言をいただいたり，また書籍を貸して頂きました。

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ の作者である Leslie Lamport 氏， $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ の開発をされた Frank Mittelbach 氏，Johannes Braams 氏，David Carlisle 氏，Michael Downes 氏，Alan Jeffrey 氏，Sebastian Rahtz 氏，Chris Rowley 氏，Rainer Schöpf 氏， $\text{T}_{\text{E}}\text{X}$ 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 $\text{p}_{\text{M}}\text{T}_{\text{E}}\text{X}$ を移植して下さった角藤亮氏， dviout を開発された大島利雄氏と乙部巖己氏， $\text{B}_{\text{I}}\text{T}_{\text{E}}\text{X}$ 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趙珍煥氏，PostScript や PDF などのページ記述言語を作成された Adobe 社の方々，さらに，フリーウェア，マクロパッケージなどの作成で， $\text{T}_{\text{E}}\text{X}$ の分野において貢献されたの方々にも感謝いたします。

大友康寛氏や田中健太氏には本書の誤記を指摘していただき，さらに改善すべき箇所について議論していただきました。永田善久氏にはドイツ語表記について教えていただきました。

多くの方々が本書の作成に貢献して下さいました。本当にありがとうございます。協力して下さったの方々のためにも，本書が日本の $\text{T}_{\text{E}}\text{X}$ コミュニティにおける恒久的な財産として残り続ける事を切に望んでおります。

この本は GNU FDL で発行されています! もちろん印税免除で....

本書は GNU Free Documentation License^{*6}の書籍ですから、その原稿と PDF 版を著者のウェブページ^{*7}で公開しています。誤記・誤植や補足事項に関する情報を取り扱っています^{*8}。本書と「同じような出力を L^AT_EX で実現したい」と感じたのであれば、直接原稿を参照してみてください。

本書の印刷用の PDF, hyperref による便利な操作が可能な閲覧用 PDF も公開しています。パソコンなどに閲覧用 PDF を保存しておけば文字列検索もできますし、紙媒体の本書がない時にも活用できるものと思います。

最後に GNU の思想^{*9}と GNU Free Documentation License を作成してくれた *Free Software Foundation* の Richard Stallman 氏 に感謝します。

^{*6} FSF によるフリーな文書の利用に関するライセンスの事です。

^{*7} <http://tex.dante.jp/typo/>

^{*8} 書名は『好き好き L^AT_EX 2_ε 初級編』という名前で公開している場合があります。

^{*9} この GNU に関する情報を公開している団体を FSF: *Free Software Foundation* (gnu@gnu.org) と言います。彼らが目指す社会、彼らの思想の詳しい事についてはウェブページ (<http://www.gnu.org/>) にアクセスすると良いでしょう。私が本書を作ったきっかけも、この FSF の活動に触発されたものです。興味がありましたらご覧ください。

目次

まえがき	i
謝辞	v
第 1 章 執筆を始める前に	1
1.1 組版とはなんだろうか	1
1.2 文章表現	1
1.3 $\text{T}_{\text{E}}\text{X}$ とは何か	3
1.4 WYSIWYG とは何か	3
1.5 一括処理とは何か	4
1.6 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ とは何か	4
1.7 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ の導入	5
第 2 章 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ の基本	7
2.1 基本の基本	7
2.1.1 処理の流れ	7
2.1.2 動かしてみる	9
2.1.3 原稿作成時の注意点	11
2.1.4 フォルダ・ファイルの基本的な操作	12
2.1.5 エラーに遭遇する	14
2.1.6 プレビューアの操作	17
2.1.7 コマンド	19
2.1.8 括弧について	20
2.2 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ に関わるファイル形式	21
2.3 コマンドの基本	23

2.3.1	原稿の先頭でのコマンド	23
2.3.2	プリアンブルでのコマンド	24
2.4	執筆環境における基本	26
2.5	原稿の出力形式	27
2.5.1	出力形式の種類の大説	27
2.5.2	L ^A T _E X の原稿から DVI へ	29
2.5.3	DVI を PDF に——Dvipdfmx	30
2.5.4	DVI を PostScript に——dvips	38
2.5.5	T _E X から HTML へ——T _E X4ht	39
第 3 章 文章の書き方		45
3.1	文章の論理構造	45
3.2	表題	46
3.3	見出し	48
3.3.1	見出しの出力	48
3.3.2	見出しの深さ	49
3.4	目次の出力	49
3.4.1	目次を出力する深さ	51
3.4.2	見出しの番号付けの深さ	51
3.5	概要の出力	52
3.6	段落と字下げ	53
3.6.1	行頭の字下げ	54
3.6.2	段落の字下げ——indent	54
3.6.3	ダブルスペース	55
3.7	長さの単位	56
3.7.1	L ^A T _E X での単位の取り決め	56
3.7.2	単位の使い方	56
3.8	句読点	57
3.9	注釈	59
3.10	文字の強調	59
3.11	そのまま出力できない記号	60
3.11.1	特殊記号	60
3.12	原稿中での空白の扱い	60

3.13	コメントの挿入	61
3.14	べた書き	62
3.14.1	特殊なべた書きその 1—alltt	62
3.14.2	特殊なべた書きその 2—cmtt	63
3.15	引用や文の区切り	63
3.15.1	書籍名や雑誌名の引用	65
3.15.2	ダッシュ	66
3.15.3	ハイフネーション	67
3.15.4	改行	68
3.16	空白の扱い	68
3.16.1	文章の中の空白	68
3.16.2	その他注意すること	70
3.16.3	和文と欧文のあいだの空白	70
3.17	行揃え	71
3.18	箇条書き	73
3.18.1	番号付箇条書き環境の拡張—enumerate	74
3.19	これまでの復習	75
3.20	書体について	76
3.20.1	文字の大きさの変更	78
3.20.2	書体の変更	79
3.20.3	基本書体の変更	81
3.21	文章の修正	81
3.21.1	構文チェック—syntonly	82
3.22	クラスとパッケージ	82
3.22.1	標準的なクラス	83
3.22.2	クラスオプション	85
3.22.3	標準で使用できるパッケージ	86

第 4 章 コマンドとマークアップ 89

4.1	マークアップ言語とは?	89
4.2	記号とコマンド	89
4.2.1	記号の分類	90
4.2.2	コマンド	91

4.2.3	コマンドの定義	92
4.2.4	文字やコマンドの区切り	95
4.2.5	コマンドの引数	100
4.3	グルーピング・入れ子構造	100
4.3.1	大域化	102
4.4	宣言と命令の違い	103
4.5	相互参照	104
4.5.1	相互参照の仕組み	105
4.5.2	カウンタ	108
4.6	相互参照の工夫	109
4.6.1	参照ラベルの表示——showkeys	112
4.6.2	相互参照に関わる L ^A T _E X の警告	112
第 5 章 数式の書き方		113
5.1	はじめに	113
5.2	数式の出力	113
5.2.1	文中数式	114
5.2.2	グルーピング	114
5.2.3	別行数式	114
5.2.4	番号付き数式	115
5.2.5	複数行数式	116
5.2.6	複数行番号付き数式	117
5.3	書体の変更	117
5.4	数式における空白の調節	119
5.5	基本的な数式コマンド	121
5.5.1	添え字	121
5.5.2	数学関数	122
5.5.3	大きさ可変の数学記号	123
5.5.4	区切り記号と括弧	124
5.5.5	行列	126
5.6	表示形式の調整	130
5.7	数式モード中の記号	132
5.7.1	ギリシャ文字	132

5.7.2	関係子や演算子などの数学記号	134
5.7.3	標準ではない数学記号—— <code>latexsym</code>	136
5.8	定義や定理など	137
5.8.1	定理型環境のカスタマイズ	138
5.9	その他有益な事柄	139
5.9.1	記号の積み重ね	141
5.9.2	記号の重ね合わせ	141
5.9.3	数式の太字	142
5.9.4	高さや幅を揃える	143
5.9.5	スマートな分数の書き方	144
5.9.6	場合分けなど	145
5.9.7	数式モード中の空白と書体	146
5.9.8	行列の省略点	147
5.10	良くある間違いと正統な入力方法	148
5.10.1	全角で数式は入力しない	148
5.10.2	文中の数式の入力	149
5.11	数式表現の拡張—— <code>AMSTeX</code>	150
5.11.1	<code>AMSTeX</code> の数式環境の概説	152
5.11.2	<code>gather</code> 環境	154
5.11.3	<code>split</code> 環境	154
5.11.4	<code>align</code> , <code>flalign</code> , <code>alignat</code> 環境	155
5.11.5	<code>multiline</code>	155
5.11.6	括弧付の行列	156
5.11.7	数式番号の工夫	158
5.11.8	ダイアグラムの例	159
5.11.9	追加された演算子等	160
5.11.10	その他のコマンド	162
5.11.11	分数の拡張	164
5.11.12	<code>AMSFonTS</code> の数学記号	165
5.12	<code>txfonts</code> / <code>pxfonts</code> での拡張	165

第 6 章 図表の構成 171

6.1	一般的な取り決め	171
-----	--------------------	-----

6.2	L ^A T _E X での扱い	172
6.3	表	173
6.3.1	表中の脚注	177
6.3.2	tabular 環境の出力位置	177
6.3.3	書籍スタイルの表罫線——booktabs	178
6.3.4	小数点揃え——dcolumn	179
6.3.5	表における行の連結——multirow	180
6.3.6	ページを跨ぐ表——longtable	181
6.3.7	表の幅の指定——tabularx	184
6.3.8	表作成支援ツール	185
6.4	図に関する制約と画像の扱い	186
6.5	画像ファイルの張り込み	187
6.5.1	デバイスドライバの選択	187
6.5.2	具体的な手順	188
6.5.3	張り込みにおけるオプション	190
6.5.4	画像の拡大や回転等の操作	192
6.5.5	Dvipdfmx における EPS 画像の扱い	193
6.5.6	PDF 画像の切り抜きと BoundingBox	194
6.5.7	dvips と Dvipdfmx の併用	196
6.5.8	レポート・論文における図の張り込み	196
6.5.9	汎用的な画像の作成と活用	197
6.5.10	プログラム特有の処理	197
6.6	図の張り込みの際の工夫	200
6.6.1	図を二つ横に並べる	200
6.6.2	画像に文字を追加する——labelfig	201
6.7	描画の方法	203
6.7.1	ベタ書きによる図の作成	203
6.7.2	曲線の描画	204
6.7.3	picture 環境による描画	205
6.7.4	picture 環境の拡張その 1——epic	208
6.7.5	picture 環境の拡張その 2——eepic	209
6.7.6	picture 環境の拡張その 3——pict2e	211
6.8	他のプログラムによる描画	211

6.8.1	TPIC による描画	212
6.8.2	メタな描画プログラム	213
6.8.3	PSTricks	215
6.8.4	Xy-pic	217
6.8.5	化学式・化学構造式	217
6.8.6	グラフの描画	217
第 7 章 文献一覧の作成		219
7.1	参考文献の明記	219
7.2	参考文献を手動で並べる場合	220
7.2.1	文献の並べ方	221
7.3	参考文献をプログラムで並べ替えるとき	222
7.3.1	jBIBTEX の使い方	222
7.3.2	文献データベースの作成	223
7.3.3	参考文献一覧の出力	224
7.3.4	文献の種類及び項目	228
7.3.5	各文献スタイルの出力例	230
7.3.6	文献の追加例	232
7.3.7	文献の複数参照——cite	234
7.3.8	参照の形式を変更する	235
7.4	文献の管理	236
第 8 章 L^AT_EX の応用		237
8.1	ページレイアウトの簡単な設定	237
8.1.1	版面のレイアウト	237
8.1.2	簡単なページレイアウト——geometry	240
8.1.3	ヘッダやフッタの設定その 1	249
8.1.4	ヘッダ・フッタの変更その 2——fancyhdr	250
8.1.5	ページ/総ページ	253
8.2	レイアウトの制御	253
8.3	その他のコマンド	254
8.3.1	日付	254
8.3.2	L ^A T _E X のロゴ	255

8.4	単位・通貨の出力について	255
8.5	あらかじめ定義されている見出しの変更	257
8.6	目次再び	257
8.7	多段組	259
8.7.1	3 段組以上の多段組——multicol	260
8.8	長さ	263
8.9	箱の操作	265
8.9.1	枠のない箱	265
8.9.2	枠のある箱	266
8.9.3	広範囲な箱	267
8.9.4	箱の保存と使用	268
8.9.5	箱の上げ下げ	269
8.9.6	罫線と下線	269
8.9.7	枠付きの箱その 2——fancybox	270
8.10	空白の挿入	274
8.10.1	水平方向の空き	274
8.10.2	垂直方向の空き	275
8.11	伸縮する糊	276
8.11.1	空白ではないグルー	277
8.12	付録の追加	278
8.13	原稿を複数のファイルに分ける	279
8.14	翻訳作業	280
8.15	用語の統一	280
8.16	色の指定——color	281
8.16.1	要素に色を付ける	281
8.17	プログラムソースの挿入——listings	284
8.17.1	言語の設定	286
8.17.2	主な設定値	288
8.18	URL の記述——url	292
8.19	ハイパーリンクの実現——hyperref	292
8.19.1	パッケージオプション	293
8.20	原稿の執筆支援	296
8.20.1	入力支援統合環境——YaTeX	296

8.20.2	原稿のコンパイル支援——Make	296
8.20.3	latexmk	306
8.20.4	原稿の版管理——CVS	306
8.21	簡単な計算と条件分岐	307
8.21.1	簡単な四則演算——calc	307
8.21.2	条件分岐——ifthen	309
第 9 章 文書のサンプル		313
9.1	投稿・概要論文のサンプル	313
9.2	学位論文のサンプル	318
9.2.1	クラスファイルが提供されている場合	318
9.2.2	クラスファイルが提供されていない場合	320
付録 A 最近の動向		327
A.1	PDF と T _E X	327
A.2	文字と書体	328
A.2.1	日本語とユニコード周辺	332
A.3	日本語クラスファイル	333
A.4	画像やグラフィックス周辺	333
A.5	今後について	333
A.5.1	ptetex	334
A.6	環境依存の話	336
A.6.1	Vine Linux	336
付録 B 参考資料		339
B.1	L ^A T _E X と直接関係のない参考資料	339
B.2	L ^A T _E X の書籍	342
B.2.1	入門書その 1	342
B.2.2	入門書その 2	343
B.2.3	数学系	343
B.2.4	化学・生化学	344
B.2.5	マクロやクラスの作成	344
B.2.6	T _E X についての本	344

B.3	文書作成全般	345
B.3.1	作文技術	345
B.3.2	組版全般	345
B.3.3	多少入手が難しい書籍	345
B.3.4	無料の冊子	346
B.4	ウェブの資料	346
B.4.1	CTAN と Ring Server の使い方	346
B.4.2	L ^A T _E X	347
B.4.3	L ^A T _E X 周辺の資料	349
B.4.4	マクロパッケージ	350
B.5	ウェブページ	351
付録 C 変更履歴		353
付録 D The GNU Free Documentation License		355
D.1	Preamble	355
D.2	Applicability and definitions	355
D.3	Verbatim copying	356
D.4	Copying in quantity	356
D.5	Modifications	357
D.6	Combining documents	358
D.7	Collections of documents	358
D.8	Aggregation with independent works	359
D.9	Translation	359
D.10	Termination	359
D.11	Future revisions of this license	359
命令索引		361
索引		372

図目次

2.1	処理の流れ	7
2.2	T _E X4ht の動作の概要	40
3.1	テキスト入力の出力例	77
6.1	longtable の使用例の出力結果	183
6.2	tabularx 使用例の出力結果	185
6.3	1 段組で横に図を二つ並べる	200
6.4	labelfig の使い方	203
6.5	制御点と式から得られるベジェ曲線	205
6.6	TPIC の使用例	212
6.7	METAPOST の出力例	215
8.1	版面のレイアウトに使用できる長さ	238
8.2	multicol の使用例の出力結果	264
A.1	基本書体の変更例	329
B.1	Ring Server の探検	348

表目次

2.1	Windows OS の基本コマンド	12
2.2	Unix 系 OS の基本コマンド	13
2.3	Dvipdfm _x でのセキュリティレベルの指定	33
3.1	文書の構成要素	46
3.2	L ^A T _E X での見出しの定義	48
3.3	見出しの階層	49
3.4	L ^A T _E X で使用できる主な単位	56
3.5	SI の基本単位	57

3.6	10^n の修飾子	57
3.7	特殊記号	61
3.8	アクセント記号	61
3.9	ダッシュなど	67
3.10	揃えの命令と宣言	72
3.11	文字の大きさの変更	78
3.12	基準の文字の大きさによるコマンドの挙動の違い	79
3.13	書体を変更するコマンド	80
3.14	和文書体のファミリー	81
4.1	カテゴリーコードの一覧	99
4.2	あらかじめ定義されているカウンタ名	106
4.3	要素に応じたラベルの貼り方	109
5.1	数式モードにおける書体の変更	118
5.2	amssymb による数式書体の拡張	118
5.3	数式における空白の制御	120
5.4	添え字の使い方の例	121
5.5	主な数学関数	122
5.6	大きさ可変の数学記号	123
5.7	主な区切り記号	125
5.8	括弧の大きさを指定する例	125
5.9	array 環境の主な列指定子	127
5.10	array 環境中での罫線の命令	128
5.11	数式の表示形式の変更	130
5.12	ギリシャ小文字	132
5.13	ギリシャ小文字の変体文字	133
5.14	ギリシャ大文字	133
5.15	関係子	134
5.16	二項演算子	134
5.17	大型演算子	134
5.18	小さいアクセント	135
5.19	大きいアクセント	135

5.20	矢印	135
5.21	特殊な数学記号	136
5.22	点	136
5.23	標準ではない数学記号	137
5.24	amsmath で追加されたギリシャ大文字の変体文字	160
5.25	amsmath で追加された数学関数	160
5.26	amsmath で追加された積分記号	161
5.27	amsmath で追加されたアクセント記号	161
5.28	amsxtra で追加された添字アクセント記号	161
5.29	amsmath で追加された空白命令	162
5.30	$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$ の二項演算子	165
5.31	$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$ の二項関係子	166
5.32	$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$ の否定二項関係子	167
5.33	$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$ の矢印記号	167
5.34	$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$ の否定矢印記号	167
5.35	$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$ のギリシャ文字とヘブライ文字	168
5.36	$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$ の区切り記号	168
5.37	その他の $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$ 数式記号	168
5.38	その他の文字記号	168
5.39	txfonts/pxfonts で拡張された二項演算子	168
5.40	txfonts/pxfonts で拡張された数学記号	169
5.41	txfonts/pxfonts で拡張された大型演算子	169
5.42	txfonts/pxfonts で拡張された区切り記号	169
5.43	txfonts/pxfonts での変体文字	169
5.44	txfonts/pxfonts で拡張された二項関係子	170
6.1	浮動体の種類	172
6.2	浮動体の位置指定	173
6.3	tabular 環境の主な列指定子	174
6.4	tabular 環境中での罫線の命令	175
6.5	表の出力例	176
6.6	各種デバイスドライバの画像形式対応状況	187

7.1	文献の形式	229
7.2	フィールド名	230
7.3	文献の種類における必須・任意項目	231
7.4	cite パッケージで変更できる命令	235
8.1	ヘッダやフッタの指定	250
8.2	ページ番号の種類指定	251
8.3	textcomp で使える記号	258
8.4	定義済みの見出しの変更	259
8.5	L ^A T _E X の歴史	272
8.6	改行を許す水平方向の空き	274
8.7	改行を許さない水平方向の空き	275
8.8	垂直方向の空き	276
8.9	L ^A T _E X で使用できるグルー	277
8.10	主なリーダー	278
8.11	listings で使用できる主な言語	287
A.1	フォント関連のパッケージ一覧	328

第 1 章

執筆を始める前に

L^AT_EX というプログラムと文章を紙面に構成する上で重要となる組版というものについて少し紹介します。また、L^AT_EX の歴史的背景と諸事情についても簡単に触れておきます。

1.1 組版とはなんだろうか

組版^{くみはん}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事です。

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は一定のルールに沿って組版されているものです。例えば 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の組織が各々で定めています。

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せらう、読者を相手にしている事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。

1.2 文章表現

L^AT_EX を用いるとユーザ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). His famous equation \$ E = mc^2 \$ is written in the theory.

“‘Stop!’ the man said.” \par 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（paren）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。‘fig.’ と ‘3’ のあいだで改行する事は好ましくないので、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事になります。

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

agenda ← *office*

agenda ← *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」という事になりますが、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（と組版）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」が相手に伝わる事になります。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事が難しくなります。

本書でもそのような「記号の使い方」に関する部分を取り扱い、それらを L^AT_EX 上でどのように実現すれば良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 L^AT_EX 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 の Writer や Microsoft Office の Word などがある類です。これらのワープロソフトと L^AT_EX のあいだ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、‘I’ という文字をワープロソフトで斜体 (*I*) 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。L^AT_EX をうまく使いこなせば、視覚的にその文字の意味を認識しながら、文書を執筆する事ができるようになります。

1.3 T_EX とは何か

テック
T_EX [53] とは Donald Knuth 氏によって開発された組版プログラムです。特筆すべき点は数式の処理に優れている事、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る機能を持っている事などです。

この T_EX を用いれば書式が統一されてより美しい文書を作成する事ができます。オフィスソフトよりも論理構造のきちんとした文章が作成できます。無論 T_EX がそれらの処理を自動でやってくれるわけではないので、ユーザが必要な命令を明示的に記述します。覚えるまでに多少時間がかかるのが難点ですが、慣れれば文書の編集には便利なツールです。

1.4 WYSIWYG とは何か

ワイジイワイグ
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事を言います。T_EX 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。

1.5 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。これは当然の事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事で、仕上がりは**全てのページの組版が終了するまで**分からないという事です。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。

1.6 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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も手続きが非常に多いようです。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事ができます。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 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ と言います。

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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、次のような記述があるとします。

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

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事を行います。

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

HTML での表記が $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ではこのようになるので、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。

$\text{T}_{\text{E}}\text{X}$ も $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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事ができませんが、中野賢氏を始めとするアスキーの方々が $\text{T}_{\text{E}}\text{X}$ の日本語化をし

てくださいましたので、今ではこの T_EX/L^AT_EX 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 T_EX や L^AT_EX をそれぞれ pT_EX, pL^AT_EX と呼びます。

L^AT_EX が最初に登場したときのバージョンがあり、この頃のものを L^AT_EX 2.09 と区別しています。それから煩雑であった L^AT_EX 2.09 システムを整備して L^AT_EX 2_ε が L^AT_EX プロジェクトチームによってリリースされました。次の L^AT_EX のバージョンは L^AT_EX 3 と呼ばれていますが、このバージョンが登場するのはもう少し先のようにです。

1.7 L^AT_EX の導入

L^AT_EX の導入に関しては可能であれば近くにいる詳しい方にインストール方法を聞いて導入した方が無難です。もし個人的に導入するのであれば、環境によって次のようにインストールする事になります。なるべくウェブから最新版の T_EX 環境を導入するようにし、可能であれば定期的に更新してください*1。

Windows 阿部紀行氏*2による『T_EX インストーラ 3』を用いると非常に簡単に T_EX に関わるソフトウェア（角藤版 T_EX, dviout, Ghostscript, GSView, jsclasses）を導入する事ができます。このインストーラについては、例えば大友康寛氏による『ワープロユーザーのための L^AT_EX 入門』にあるインストールの解説*3を参照してみてください。

Mac OS X MacOS X WorkShop*4で簡単に周辺ツールも導入できます*5。今後の展開については MacWiki*6等を参照してください。

Vine Linux コンソールから管理者権限で `apt-get install task-tetex` と実行するだけで T_EX 関係のパッケージが導入されます。

Fedora Core 土村展之氏による ptetex3 において Fedora Core 5 用の RPM が提供されています*7。

*1 現在 T_EX のシステムを更新するための指針が提示されていますが、本書では詳しく扱いません。簡単に説明すると複数の texmf ツリーと呼ばれるディレクトリを用意し、配布されている T_EX のファイルと自分が後から追加したファイルを分離する、というような事が可能になる管理方法があります。

*2 <http://www.ms.u-tokyo.ac.jp/~abenori/>

*3 <http://www.klavis.info/texinstnew.html>

*4 <http://www.bach-phys.ritsumei.ac.jp/OSXWS/>

*5 X11 も導入していれば、GUI インタフェースの Synaptic によるパッケージ管理も可能となります。

*6 <http://macwiki.sourceforge.jp/>

*7 <http://tutimura.ath.cx/~nob/tex/ptetex/ptetex3/rpm/>

LaTeX の導入と周辺情報に関しては奥村晴彦氏による TeX Wiki^{*8}を参照するのが良いと思います。数ヶ月古いというだけで何らかの問題が発生する可能性もありますから、可能な限りインターネットから最新の LaTeX を導入するようにしてください^{*9}。

Emacs のようなテキストエディッタやコンソールからの操作等に慣れていない方は、TeX 環境とは別に、TeX の執筆支援環境も導入すると大変便利かと思われれます。2.4 節を参照し、それぞれの環境に応じて適切だと思うプログラムを導入してみてください。

^{*8} <http://oku.edu.mie-u.ac.jp/~okumura/texwiki/>

^{*9} 書籍の付録として TeX 環境を提供すると、逆にユーザにトラブルの原因を増やしてしまいかねないため、本書にはそのような類いのものは付与しません。

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

L^AT_EX は普通のワープロとは違い、ある程度の基本的な前提事項を踏まえなければなりません。ここでは L^AT_EX を操作する上での基本の基本を解説します。

▼ 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事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

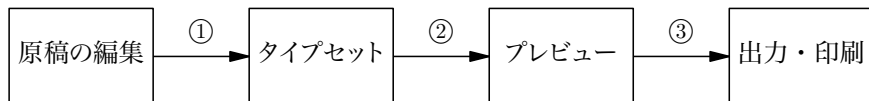


図 2.1 処理の流れ

ありますが、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 (最近では L^AT_EX ファイルから直接 PDF を作るプログラム pdfL^AT_EX など存在します。2006 年 4 月現在日本語化はされていませんが、将来的には日本語を含む文書でも pdfL^AT_EX が主流になると思われます)。

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**と呼びます。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 (T_EX/L^AT_EX は直接コンソールからプログラムを呼び出し、文書を執筆する事も可能です)。Unix 系 OS ではコンソールから次のように入力すると Emacs が立ち上がると思います。

```
$ emacs filename.tex &
```

Emacs でなくとも、お好きなテキストエディッタや、何らかの 8.20 節にあるような執筆環境が良いと思います。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 filename.tex
```

とすれば文書が成形されます。この作業の事を良く**タイプセット**とか**コンパイル**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 filename.tex そのものが整形されるわけではなく新たに filename.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事を**プレビュー**すると言います。Unix 系 OS ならば

```
$ xdvi filename.dvi &
```

などとすると良いでしょう。最後のアンパサンド '&' 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事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。「L^AT_EX をやりたいんですが」と言い出せば大抵の L^AT_EX ユーザは親切に対応してくれると思います。

とりあえず自分のいつも使っているテキストエディタ（メモ帳や Emacs 等）で以下のようなファイル `test.tex` を作成してください^{*1}。Windows ユーザの方はバックslash ‘\’ を円記号 ‘¥’ に読み替えてください。

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

Unix 系 OS ならば Emacs 等で良いでしょう。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には

```
[_]J.:---Emacs: test.tex (LaTeX)-[L1-A11----]
```

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]` の表示ですと日本語が入力できる状態です。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^{*3}。詳しくはご自分の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（L^AT_EX 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

^{*1} 無用なトラブルを避けるため、ファイル名には半角英数の文字だけを使い、可能な限り日本語は使わないようにしてください。

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

^{*3} 比較的新しいと思われる Unix 系 OS であれば「全角/半角」入力の切り替えが `半角/全角` キーでできる場合もあります。

「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すが、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 test.tex
```

としてタイプセットしてください。するとコンソール（端末）には次のように表示されるとおもいます。

```
┌
  This is pTeX, Version 3.14159-p3.1.5 (euc) (Web2C 7.4.5)
  (./test.tex
  pLaTeX2e <2005/01/04>+0 (based on LaTeX2e <2001/06/01> patch level
    0)
  (/usr/local/share/texmf/ptex/platex/base/jarticle.cls
  Document Class: jarticle 2002/04/09 v1.4 Standard pLaTeX class
  (/usr/local/share/texmf/ptex/platex/base/jsize10.clo))
  No file test.aux.
  [1] (./test.aux) )
  Output written on test.dvi (1 page, 236 bytes).
  Transcript written on test.log.
└
```

始めにバージョン情報を表示^{*4}して終わりには test.dvi に組版後のファイルを出力し、処理状況を test.log に書き出した事になっています。test.tex をタイプセットして出力された test.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。



T_EX/L^AT_EX がコンソールに表示するメッセージ（及び $\langle file \rangle$.log）には、様々な情報が出力されます。丸括弧の後にファイル名があるような場合、上記の例では (./test.tex というのは、ファイルが読み込まれている階層を表しています。内容をもう少し分かりやすく書けば次のようになります。

```
(./test.tex
 (/usr/local/share/texmf/ptex/platex/base/jarticle.cls
  (/usr/local/share/texmf/ptex/platex/base/jsize10.clo)
 )
 (./test.aux)
 )
```

test.tex が jarticle.cls と test.aux を子ファイル、jarticle.cls が jsize10.clo を子ファイルとしてファイル読み込みを行っている事が分かります。

^{*4} 実は T_EX のバージョンは Donald Knuth 氏の思想により 3.14159... の様に π に収束するようになっていきます。

各括弧と数字の場合、例えば‘[1]’はページ番号を表しています。コンソールに‘[’のような表示で L^AT_EX の処理が中断したとすれば、あるページを組み上げている段階で何らかの問題に遭遇した事を意味します。

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。コンソールから ls (Windows の方は dir) コマンドで

```
$ ls test.*
```

とすると、次の四つのファイルが存在する事を確認してください。

```
┌ test.aux test.dvi test.log test.tex ─┐
```

L^AT_EX の原稿であるソースファイル test.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ました、これらのファイルが一般的にどのような役割を持っているのかを説明します。

⟨file⟩.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。

⟨file⟩.dvi ⟨file⟩.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。

⟨file⟩.log ⟨file⟩.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**。

⟨file⟩.tex 先程作成した L^AT_EX 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事ができません。

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

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

となる事でしょう。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 (円) 記号を補ったり長い命令を打ち込みます。

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

`\textasciitilde`
`\textasciicircum` `~ ^ \ | < >`
`\textbackslash` `\textbar`
`\textless` `\textgreater`

▶ **問題 2.1** 半角文字と全角文字を混在させてある程度の量の文章の入力をしてください。特殊記号も含めるようにすると良いでしょう。

▼ 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
<code>mkdir</code>	新規にフォルダを作成
<code>cd</code>	ディレクトリを移動
<code>dir</code>	ファイルの情報を表示
<code>move</code>	ファイル名を変更したり移動
<code>copy</code>	ファイルをコピー
<code>del</code>	ファイルを削除
<code>help</code>	コマンドのヘルプを表示

それぞれのコマンドの使い方 (ヘルプ) は `mkdir /?` 等とする事で表示できます。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も同様の事ができます。どちらかという 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 2.4 節を参照して L^AT_EX 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かありますし、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

```
$ mkdir --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

表 2.2 Unix 系 OS の基本コマンド

```
$ man mkdir
```

とします。もっと詳しいヘルプが見たいときは

```
$ info mkdir
```

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 の基本的な操作方法、正規表現、プロセス、ファイル・ディレクトリ概念などについては付録 B の参考資料を参照してください。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。本書では 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事を**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 2.2** 以下の作業をコンソー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 ../file.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。最後の操作でうへの階層のディレクトリ（親ディレクトリ）に `file.txt` がない事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。このとき端末には疑問符 '?' が表示されます。

▷ **例題 2.3** まずは以下のソースファイル `errortest.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次のように表示されるでしょう。

```
┌────────────────────────────────────────────────────────────────────────────────┐
! Misplaced alignment tab character &.
1.3 Hello &
        Goodbye! Give me $100! Give me 100%!
?
└────────────────────────────────────────────────────────────────────────────────┘
```

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 行目でアンパサンド '&' を不正に使っていると言われていました。ここで `[Enter]` キーを押すとさらに次のように表示されます。

```

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

```

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。極め付けにもう 1 度 **Enter** キーを押すと次のような表示になります。

```

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

```

今度はドル '\$' を不正に使ったと言われました。以上の事から 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事が分かりました。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のようになるでしょう。

```

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

```

3 行目のアンパサンド '&' とドル '\$' と 4 行目のシャープ '#' 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。

```

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

```

今度はドル '\$' 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事が分かります。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。一つ目のエラーが表示されます。

```

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

```

タイプセットは中断しませんが、何か煩雑な表示が出現します。

```

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 !!$

```

これは `Overfull \hbox` という警告である事が分かります。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次のような表示がありました。

```
<inserted text> $
```

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事を意味しているようです。ですからファイル `errortest.tex` はさらに次のように修正する事になります。

```

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いきそうです。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 '?' で返事を返すと次のように表示されます。

```

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.
?

```

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は何らかの対処ができるようです。

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** キーを押し続けた事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。元の前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端末に表示します。
- X ゲームを終了します。

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から‘\end{document}’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る*5。

▼ 2.1.6 プレビューアの実行

プレビューを行うプログラムの事を**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 又は Fedora Core 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。まずターミナル上で<file>.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 <file>.dvi &
```

のようにします。Unix 系 OS ならばアンパサンド‘&’をつけて**バックグラウンド**で起動します。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。

*5 慣れないうちはどこに記述間違いがあるのかを見つけるのが難しいと思われるから、適当な場所に分かりやすいようにブレイクポイント（タイプセットが停止する地点）を設けるのも手だと思えます。

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 $\langle file \rangle$.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。



これまで操作して少し疲れたでしょうから、ここで休憩をしましょう。DVI ファイルと言われてもなじみの薄いファイル形式かもしれません。そこでこの DVI ファイルを別のファイル形式に変換してみましょう。

例えば PostScript に変換するには (Windows の方は dvipsk)

```
$ dvips -o test.ps test.dvi
```

とすると test.dvi から test.ps が生成されます。この test.ps を

```
$ ps2pdf test.ps test.pdf
```

として PDF ファイル test.pdf を生成する事もできます。Postscript ファイル test.ps をプリンタに出力するには

```
$ lpr -P  $\langle$ プリンタ名 $\rangle$  test.ps
```

とする事で印刷要求を送信できます。 \langle プリンタ名 \rangle についてはシステムの管理者に聞くなどしてください。

タイプセット時に拡張子 .tex を省略して

```
$ platex test
```

としてみてください。以上のような例でも L^AT_EX プログラムは test.tex を探し出してタ

イブセットしてくれるでしょう。

これは \LaTeX で使われているファイル検索の仕組みに関係しています。 \LaTeX では `Kpathsearch` というファイル検索機構を採用しています。まず上記の例では `test.tex` があるときに `platex test` とすると `Kpathsearch` が自動的に拡張子をつけてファイルを探します。この時どのディレクトリ(フォルダ)から探し出すかという情報が必要になります。この情報が記載されたファイルは `texmf.cnf` というファイルです。ほとんどの環境でディレクトリ `$texmf/web2c/` 以下にあります。ここで `$texmf` は \LaTeX のファイルを格納すべき一番上のディレクトリを示します。Unix 系 OS ならば `/usr/local/share/texmf/` などだったり Windows ならば `C:\usr\local\share\texmf\` だったりするでしょう。

```
$ kpsewhich texmf.cnf
```

とすると、どこに `texmf.cnf` が存在するのかが分かります。



最近の \LaTeX プログラムは `Kpathsearch` に対応しているので何も意識しなくても適切に設定ファイルやクラスファイルなどを検索してくれます。しかし手動で検索したいときもあると思います。このような場合は `kpsewhich` というプログラムを使って `less 'kpsewhich book.cls'` とすると `book.cls` というファイルが `$texmf` 以下の特定のディレクトリから検索され、`less` がファイルを整形します。実は `Kpathsearch` ではプログラム毎にファイルを検索する場所を指定できます。そのため、`jarticle.cls` というファイルを `kpsewhich jarticle.cls` としただけでは検索結果に表示されません。 `-progrname` というコマンドラインオプションを付けて `kpsewhich -progrname=platex jarticle.cls` と実行すると、 \pTeX が `jarticle.cls` というファイルを見つけられる事になります。これにより複数のプログラムで同じ名前のファイル名が存在しても、ディレクトリを変えておけば衝突しませんし、ファイルを検索するディレクトリが少なくて済みますので処理速度が向上します。

▼ 2.1.7 コマンド

\LaTeX では原稿を三つのパートに分割する事ができます。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を記述)

```
\documentclass[<オプション,...>]{<クラス>}[<リリース>]
```

<前書き部分> (プリアンブルコマンドを記述)

```
\begin{document}
```

<本文> (ボディ)

```
\end{document}
```

この中で `\documentclass`, `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

事ができ、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事ができます。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事ができないように 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ましょう。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、**本書では**以下のように取り決めます*6。

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。引数を取る事ができる。

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。引数を取る事ができる。

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{見出し語}`

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]{クラス名}`

▼ 2.1.8 括弧について

さて、L^AT_EX の基本を知った所で**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。

*6 コマンドについての説明としては不十分なのですが、今は命令と環境があると解釈してください。

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

上記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。主に単一ページのベクトル画像などに使われる。
- .ist 索引の書式を決めるファイル。索引スタイルと言う。

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。
- .bbl 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。
- .idx 並べ替えられる前の索引の語句が書き出されたファイル。MakeIndex, mendex などのプログラムで並べ替えをする。
- .ind makeindex などによって並べ替えられた索引ファイル。標準的には theindex 環境を用いて記述されている。
- .ilg makeindex などを実行したときの処理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、主に以下のものがあります。

- .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 標準の無圧縮ビットマップ画像。
- .png 可逆圧縮で Dvipdfm が標準で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 CreateBB 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 原稿の先頭でのコマンド

少し前置きが長くなりましたが L^AT_EX の原稿の構造をもう一度見ておきましょう。

```
\documentclass[〈オプション,...〉]{〈クラス〉}[〈リリース〉]  
〈前書き部分〉 (プリアンブルコマンド)  
\begin{document}  
〈本文〉 (ボディ)  
\end{document}
```

さて、原稿の先頭部分、`\documentclass` が始まる前のイニシャルコマンドには `filecontents` 環境が使えます。

```
\begin{filecontents}{〈ファイル名〉}  
〈内容〉  
\end{filecontents}
```

この `filecontents` 環境が持つ機能ですが、指定した〈ファイル名〉に〈内容〉を書き出してくれます。例えば原稿を一つのファイルとしてしか配布できない場合に EPS 画像などを同時に含めるならば、この部分に EPS 画像のソースを記述します。ただしこの環境は書き出すファイルの先頭にコメントを挿入しますので、アスタリスク ‘*’ を付けると自動的に付加される余分なコメントが入りません。

```
\begin{filecontents*}{〈ファイル名〉}  
%!PS-Adobe-2.0なんかかかんとか...
```

```
\end{filecontents*}
```

▼ 2.3.2 プリアンブルでのコマンド

原稿の先頭には `filecontents` 環境が使えることは分かりました。次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「これから文書で使う命令の定義や前書きを書きます」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 3.22.1 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できます。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、L^AT_EX はタイプセット時に次のような警告 (warning) を出します。

```
LaTeX Warning: You have requested, on input line 1, version
'2003/12/31' of document class jarticle,
but only version
'2002/04/09 v1.4 Standard pLaTeX class'
is available.
```

他にも 3.22.3 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 L^AT_EX で扱

いたいと思い、デバイスドライバとして `Dvipdfmx` を使う場合は次のように `graphicx` パッケージを使うことを**プリアンブル**で宣言します。

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

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**。

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

例えば `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` などを次のように宣言できますが、そうすると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。

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

基本的なソースファイルの構成は次のようになります。

```
\documentclass[a4j]{jarticle}
\usepackage[dvipdfmx]{graphicx}
\usepackage[dvipdfmx,usenames]{color}
\begin{document}
ここに文章を記述します。
\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[dvipdfmx,a4j]{jarticle}
\usepackage{graphicx}
\usepackage[usenames]{color}
```

これによりドキュメントクラスオプションが**グローバルオプション**としての機能を果たし、`\usepackage` で読み込まれるマクロパッケージ全てに渡される事になります。

ET_X 処理を実行した原稿のプリアンブルに以下を記述します。

```
\listfiles
```

`\listfiles` 命令を記述すれば、自分が処理している原稿に何のファイルが使用されているのかを端末と `(file).log` に書き出します。

▶ **問題 2.4** 実際に以下のファイル `listfile.tex` を作成、タイプセットしてください。

```
\documentclass{jbook}
\listfiles
\begin{document}
  test
\end{document}
```

出力結果からどのような事が分かるでしょうか。ここで少し疑問に思っていたきたいことは、「原稿には `jbook` を使うことしか宣言していないのに何か別のファイルも一緒に読み込まれている」ということです。この例では `pldefs.ltx` をはじめとして、`kinsoku.tex` や `jsize10.clo` 等のファイルが読み込まれています。

2.4 執筆環境における基本

T_EX はテキストエディタによって原稿を執筆するという方法を取るため、何らかの執筆環境を必要とします。それらの執筆環境の中には作業の簡略化を目的としたものも数多くあります。T_EX における伝統的な (obsolete) 執筆環境には次のようなものが挙げられます。

Unix 系 OS T_EX とその周辺プログラムを活用しようと思えば、Unix 系 OS を使うと (人によっては) 快適な執筆環境を得る事ができます。Vine Linux は特に T_EX 周辺の日本語環境が整っていると思われます*7。

Emacs L^AT_EX の原稿となるソースファイルを編集する時に役に立つテキストエディタです。

YaT_EX 上記 Emacs 上で動作する広瀬雄二氏*8による L^AT_EX 執筆支援システムです。

Tgif Unix 系 OS で広く使われているベクター画像編集プログラムです。

Gnuplot Unix 系 OS で広く使われているグラフを描画したり、データをプロットするためのプログラムです。

*7 <http://www.vinelinux.org/>

*8 <http://www.yatex.org/>

Make 原稿の再コンパイルを支援するためのプログラムです。Makefile という特別なファイルを用意する事で、再コンパイルにおける手間を軽減する事になります。

環境に依存してはいるものの、以下に挙げるように L^AT_EX での煩雑な作業を軽減できる有益な原稿執筆支援環境が数多く存在します。

EasyT_EX 中川仁氏による Windows 用の執筆支援環境です^{*9}。L^AT_EX に慣れないうちは EasyT_EX を使うのが望ましいでしょう。導入方法や操作方法に関しては大友康寛氏による解説^{*10}や T_EX Wiki^{*11}等を参照してください。

T_EXShop Mac OS X で使用できる Richard Koch 氏らによる執筆支援環境です^{*12}。PDF でのプレビューが可能でディスプレイにおける表示がきれいです。

基本的にフリーウェアで済ませたいので、上記のような選択肢になる人も多い事でしょう。EasyT_EX や T_EXShop ではコマンドの入力を補完したり、プログラムの実行等も簡単にできる環境が整備されています。まず最初はこのようなプログラムを使った執筆の方が負荷も少ないと思われます。

もちろん、シェアウェアの方がサポートもありますし、バージョンアップも確実な部分があると思います。いずれにしてもその人にとって適切だと思われるツールは多少なりとも異なると思われますので、いくつか試用してみてください。

2.5 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事ですが、どのようなファイル形式にするかは用途によって分かれるところです。この節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。

▼ 2.5.1 出力形式の種類概説

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事ですが、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。

^{*9} <http://www.juen.ac.jp/math/nakagawa/nakagawa.html>

^{*10} <http://www.klavis.info/etexinst.html>

^{*11} <http://cise.edu.mie-u.ac.jp/~okumura/texwiki/?EasyTeX>

^{*12} <http://www.uoregon.edu/~koch/texshop/>

目的と気分によってその形式を変えますが、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事ができます。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 と書く事がありますし、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。こちらは単一ページ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。2006 年 4 月現在の最新バージョンは 1.6 で、プレビューと印刷結果が同程度の品質を得る事ができる形式です。互換性を考慮すればバージョンは 1.3 で統一するのが無難だと思われま。PDF は世界中で広く使われています。2006 年 4 月現在で日本語化はされていませんが、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。現在広く用いられているのは PDF 形式ですから、本書でも PDF とその周辺に関して詳しく解説します。

▼ 2.5.2 L^AT_EX の原稿から DVI へ

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 の原稿をコンソールから次のようにすれば、L^AT_EX の原稿ファイル $\langle filename \rangle.tex$ から DVI ファイル $\langle filename \rangle.dvi$ が生成されます。

```
$ platex filename.tex
```

このとき通常はアスキーによって日本語化された pL^AT_EX を用います*13。

互換性の為に、古い L^AT_EX, L^AT_EX 2.09 時代のソースファイルをタイプセットするには platex209 コマンドを使います。

```
$ platex209 oldfile.tex
```

学会等によっては L^AT_EX 2_ε に対応していない古い書式のクラスファイルやスタイルファイルしか提供していない場合があります。L^AT_EX 2_ε と L^AT_EX 2.09 を見分ける方法は簡単です。L^AT_EX の原稿 $\langle file \rangle.tex$ の先頭の命令に注目します。

- `\documentclass` 命令を使っていれば L^AT_EX 2_ε 用のファイル。
- `\documentstyle` 命令を使っていれば L^AT_EX 2.09 用のファイル。

L^AT_EX 2.09 時代の場合は、`\usepackage` 命令は使えません。そのため、`\documentstyle` の任意引数に必要とするスタイルファイルを列挙します。

```
\documentstyle[url,mysetting,...]{jarticle}
```

話を戻してタイプセット後に整形される 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**。

Windows では大島利雄氏らが開発している dviout , Unix 系 OS ならば xdvi, Red Hat や Fedora Core では pxdvi が使えます。Mac OS X では内山孝憲氏による Mxdvi でプレビューできます。

DVI ファイルから印刷ができるか、画像が表示できるか、どの画像形式に対応しているかというような条件は全てお使いの環境のデバイスドライバに依存しています。デ

*13 アスキーのプログラムとは別に NTT によって日本語化された jL^AT_EX も存在します。

バイスドライバの設定方法, 基本的な操作方法等は, 各種お使いのデバイスドライバのマニュアルを参照してください.

▼ 2.5.3 DVI を PDF に——Dvipdfm x

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, パソコンの画面においても印刷したのと寸分違わぬ表示を得る事ができます. 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. PDF ファイルを閲覧するには多くの環境において使用可能な Adobe Reader が利用できます. 他にも Windows では Foxit Software Company による Foxit Reader, Mac OS X ならば標準付属のプレビュー (切り抜きなどの簡単な編集も可能), Unix 系 OS であれば Xpdf などがあります.

Mark Wicks 氏が作成した Dvipdfm [86] を使うと DVI ファイルから PDF を作成できます. 平田俊作氏の日本語化パッチを当てたバージョンがそれぞれの環境で入手できます. それから現在 Dvipdfm は平田俊作氏と趙珍煥氏が中心となって活動している Dvipdfm x Project Team によってさらに改良が加えられ Dvipdfm x へと進化しています. Dvipdfm は少々古くなっていますので, 後継の Dvipdfm x を使う事をお勧めします.

Dvipdfm x は主に PDF ブックマーク, HyperT_EX, T_PIC スペシャルなどの機能をサポートしています. 画像ファイルは JPEG, PNG, EPS, EPDF, BMP (BMP は 2005 年 8 月に対応) 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されれば, そのまま PDF に取り込む事ができるようになります.

Dvipdfm x ではコマンドラインオプションによって出力結果に対する細部の調整を行う事ができます. Dvipdfm と共通なオプションは以下の通りです.

- c カラースペシャルを全て無効にします. 白黒印刷のときなどに使います.
- f <ファイル名> フォントマップファイルを指定します.
- m <数字> ページの拡大率を指定します. -p オプションと併用すると良いでしょう.
- 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. このようにしなくとも原稿のプリアンブルで次のようにしても同じ結果になります.

```
\AtBeginDvi{\special{pdf:papersize width 210mm height 270mm}}
```

jsclasses ではドキュメントクラスオプションに *papersize* を指定するだけで同様の効果を得る事ができます。

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

- l 用紙を横置きにします。ソースファイル中でドキュメントクラスオプションの *landscape* が有効でなければ意味がありません。
- 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、コン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事もできます。また悪ふざけで '-s -,-' とするとどのような出力になるか試してみると良いでしょう。
- r (解像度) PDF ファイルの解像度を指定します。標準は 600 dpi 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事があります。互換性を優先しなければならないときなどに使います。
- x (長さ) 水平方向のオフセットを指定します。標準は 1.0in です。単位には mm, cm, in, pt が使えます。
- y (長さ) 垂直方向のオフセットを指定します。標準は 1.0in です。単位については -x と同様です。
- 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 ですのでビットマップ画像などの画質を落とさたくない場合は 0 などにするとう良いでしょう。
- v 処理内容を標準出力に詳しく表示します。通常ならば、標準エラー出力に結果が表示されます。これをファイルに保存したければリダイレクトの前に 2 を付け加えて次のように実行します。

```
$ dvi2pdf -v file.dvi 2>file.pdf
```
- vv さらに処理内容を詳しく表示します。

白黒印刷用の 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次のようにします。

```
$ dvi2pdf -c -s 15-20 -o output.pdf input.dvi
```

入力ファイルの拡張子 `.dvi` は次のように省略しても構いません。

```
$ dvipdfmx input
```

PDF ファイルを Adobe Reader や Acrobat Reader などでも閲覧しているときに `Dvipdfmx` による DVI ファイルの変換を行うと `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。

`Dvipdfmx`^{*14}は中国語 (Chinese), 日本語 (Japanese), 韓国語 (Korean), 16 ビットエンコーディングの文字コード (Unicode など) にも対応しています。CID フォントの埋め込みによって日本語フォントなどを持っていない人でも日本語 PDF を表示できるようになっています。PDF のセキュリティ機能も使う事ができます。基本的に `Dvipdfm` の上位互換なので `Dvipdfm` で可能な事は `Dvipdfmx` でも可能です^{*15}。

`Dvipdfmx` で指定できる主なコマンドラインオプションは以下の通りです。

- S PDF のセキュリティを有効にします。
- K (数字) PDF のセキュリティのキービットを指定します。40 か 128 です。標準で 40 です。
- P PDF のセキュリティのレベルを設定します。
- p (幅), (高さ) 定義済みの 'a4' 以外にも、用紙のサイズを単位付きで '20cm, 20cm' のように指定する事もできます。

`Dvipdfmx` の `-P` オプションによる PDF のセキュリティの設定については表 2.3 を見てください。

0x04 から 0x20 までのビットにそれぞれ許可・不許可が割り当てられています。要は表 2.3 の 16 進数の値を 10 進数に直し、それを自分の設定したいレベルに合わせて、それぞれのビットを足したものを再び 16 進数に直せば良いのです。印刷 (0x04) と文書の改変 (0x08) だけを許可したいならばこのビットを 10 進に直して二つのビットを足します。すると 12 になるのでこれを 16 進に直してあげます。電卓などで計算すると '0x0C' になりますから `dvipdfmx -S -P 0x0C input.dvi` とすれば良い事になります。さらに `dvipdfmx -S -P 0x28 input.dvi` とすると改変と注釈の追加だけを許可するようにできますし、特に制限を課さないならば `dvipdfmx -S -P 0x3C input.dvi` とす

*14 <http://project.ktug.or.kr/dvipdfmx/>

*15 唯一フォントライセンスやファイルサイズ等の問題により `-e` コマンドラインオプションが削除されています。

表 2.3 Dvipdfmx でのセキュリティレベルの指定

ビット	印刷	改変	文字列などのコピー	注釈の追加
0x04	許可			
0x08		許可		
0x10			許可	
0x20				許可
0x28		許可		許可
0x3C	許可	許可	許可	許可

るとパスワードによる保護と暗号化のみになるものと思われます。

フォントに関する設定

論文投稿や印刷所に渡すような PDF のデータを作成するときは、互換性やフォントの問題等に関して、ある程度の配慮が必要です。

自分の環境で正常に印刷できても印刷所や出版社の環境によってはフォントがない等でうまく処理できない場合があります。また低解像度のビットマップフォントが含まれている場合も受け付けてくれないかもしれません^{*16}。

日本語などのフォントを含むような原稿ですと、 $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ で処理した DVI ファイルを Dvipdfmx で PDF に変換という形が手軽な方法だと思われます。Dvipdfmx は EPS などの PostScript ファイルを画像として $\text{L}\text{A}\text{T}\text{E}\text{X}$ に張り込んでいる場合は、それらを Ghostscript の力を借りて PDF に取り込みますので Ghostscript の性能が結果に依存します。

Dvipdfmx のフォント設定ファイルは `$texmf/fonts/map/dvipdfm/base/` であるとか、`$texmf/dvipdfm/config/` 以下に `cid-x.map` という名前であります。cid-x.map は **Map ファイル** と呼ばれ、コンソールから次のようにすると Map ファイルの所在が分かります。

```
$ kpsewhich -programe=platex -expand-path='$CMAPINPUTS'
```

ファイル `cid-x.map` の中に `rml` や `gbm` という文字列が書かれた行が存在すると思

^{*16} 多くの問題は dvips で作成した PostScript ファイルを ps2pdf 等で PDF に変換した事に起因する事が多いようです。

います*17.

```
rml H Ryumin-Light
gbm H GothicBBB-Medium
rmlv V Ryumin-Light
gbmv V GothicBBB-Medium
```

それぞれ rml, H, Ryumin-Light 等は次のような意味を持っています*18.

rml/rmlv 日本語の明朝体に割り当てる書体を決めるためのラベル. rmlv は縦書き用のもの.

gbm/gbmv 日本語のゴシック体に割り当てる書体を決めるためのラベル. gbmv は縦書き用のもの.

H/V エンコーディングマップの指定. H は横書き用, V は縦書き用.

Ryumin-Light 実際に日本語の明朝体に割り当てられるフォントの名前. Dvipsd_{fm}x は Ryumin-Light*19 という名前のフォントであれば標準では PDF に対してフォントを埋め込まないようにになっています.

GothicBBB-Medium 実際に日本語のゴシック体に割り当てられるフォントの名前. GothicBBB-Medium は標準では埋め込まれません.

この記述をフォント名などに変更すると日本語のフォントに何をを使うのかが指定できます. お使いの環境の初期設定に依存するとは思いますが, 標準では日本語などのフォントを埋め込まないようになっているとします.

GRASS 国際化版 (i18n)*20 に付属する, 条件に合致すれば再配布可能である「独立行政法人 情報処理推進機構のフォント (IPA フォント)」を使う場合は次のようになります*21.

```
rml H ipam.ttf
rmlv V ipam.ttf
```

*17 Dvipsd_{fm}x のバージョンによっては別ファイルに同じような記述がある場合があります.

*18 標準的な日本語フォント設定がされているクラスファイルを使った場合に限りです.

*19 Ryumin-Light というのはモリサワから発売されている「L リュウミン L-KL」のフォント名です. GothicBBB-Medium は「M 中ゴシック BBB」に対応します. pT_EX の世界では互換性の保持や諸事情によりこの名前が使われています.

*20 <http://www.grass-japan.org/FOSS4G/readme-grass-i18n-ipafonts.eucjp.htm>

*21 IPA フォントは 2006 年 4 月現在において, 商用ディストリビューションではない Unix 系 OS で使用出来る比較的高品質な TrueType フォントです. もしも, 東風フォントやさざなみフォント, 和田研フォント等を PDF への埋め込みに使っているようでしたら, IPA フォントへ移行する事をお勧めします.


```
gbm H ipag.ttf
gbmv V ipag.ttf
```

上記の様な記述をしたファイル `ipa.map` を作成し、Map ファイルを格納すべきディレクトリに配置しておけば^{*22}、`dvipdfmx -f ipa.map file.dvi` とすると、IPA フォントを埋め込んだ PDF ファイルが作成できます。

PDF ファイルの操作

PDF ファイルは商用のプログラムを使わないと自由度の高い編集は難しいと思われまます。簡単な操作ならば Xpdf^{*23} に付属するツールを使うと良いでしょう。

下記のプログラムは PDF ファイルにセキュリティ設定がなされている場合はパスワードを必要としたり、または全く機能しない場合があります。以下のプログラムは全てコンソールから操作します。

pdftops PDF ファイルを PostScript ファイルに変換します。

pdfimages PDF ファイルに含まれるビットマップ画像を指定したディレクトリに抽出します。あらかじめ出力するディレクトリを作成しておきます。

```
$ pdfimages filename.pdf dir/
```

するとディレクトリ 'dir' に ppm 形式か pbm 形式の画像として抽出されますので、適宜お望みの変換をしてください。

pdftotext PDF ファイルの文章をテキストファイルに抽出します。フォントマップファイルを必要とします。ASCII コード中の標準的な文字でなければうまくいかないかもしれません。

pdfinfo PDF ファイルの「文書情報」を表示します。

pdffonts PDF ファイルに使われているフォント情報を表示します。フォント名やフォントの種類、フォントが埋め込まれているかなどが分かります。

例えば、`file.pdf` という PDF が存在し、それを `pdffonts file.pdf` したとすると次のような情報が表示されます。

name	type	emb	sub	uni	object	ID
Times-Roman	Type 1	no	no	no	7	0
GothicBBB-Medium-Identity-H	CID Type 0	no	no	no	9	0

*22 配置した後に環境によっては `mktexlsr` を実行する必要があります。

*23 <http://www.foolabs.com/xpdf/>

Helvetica	Type 1	no	no	no	10	0
Ryumin-Light-Identity-H	CID Type 0	no	no	no	12	0
Times-Italic	Type 1	no	no	no	13	0
FRZWS+txsy	Type 1C	yes	yes	yes	14	0
EPSMLX+t1xtt	Type 1C	yes	yes	yes	15	0
Times-Bold	Type 1	no	no	no	16	0
LEPUME+rtxmi	Type 1C	yes	yes	yes	23	0
CACNFM+rtxsc	Type 1C	yes	yes	yes	32	0
Helvetica-Oblique	Type 1	no	no	no	65	0
UQXVYG+rtxr	Type 1C	yes	yes	yes	66	0

name PDF ファイルでのフォント名です。FRZWS+txsy とあれば、プラス + 以降が本来のフォント名になります。

type フォントの種類を表します。Type1, CID Type0, TrueType, Type1 Collection 等があります。Type3 という表示があれば、低解像度のビットマップフォントが埋め込まれている可能性がありますので、注意してください。

emb そのフォントが埋め込まれているかどうかを表します。yes であれば埋め込まれており、no であれば埋め込まれていません。

sub サブセット化されているかどうかを示します。あるフォントを PDF ファイルに埋め込むときに使っていないグリフ (字形) を埋め込まないようにします。

uni ユニコードエンコーディングされているかどうかを示します。これにより後の編集作業、文字列の抽出、テキストのコピー、文字列の検索等に影響が出る場合があります。

object ID PDF ファイルにおけるフォントの識別 ID です。

PDF の文書情報を閲覧したいときは `pdfinfo` コマンドを次のように使います。

```
$ pdfinfo file.pdf
```

すると出力結果として以下のようなものが得られます。

```
Title:          How to Write Your Own Thesis Tutorial with LaTeX2e
Subject:       For University Students and Researchers
Keywords:     TeX, LaTeX, LaTeX2e, pTeX, pLaTeX, pLaTeX2e, FUNNIST
Author:       FUNNIST
Creator:      pLaTeX2e with hyperref packages
Producer:     dvipdfmx (20040914(cvs))
CreationDate: Wed Oct 13 14:02:46 2004
Tagged:       no
Pages:        176
Encrypted:    no
```

Page size: 515.91 x 728.5 pts
File size: 1733518 bytes
Optimized: no
PDF version: 1.4

それぞれの項目の意味は次の通りです。

Title 文書の主題です。

Subject 文書の副題です。

Keywords キーワード、関連用語等です。

Author PDF の執筆者です。

Creator 元々のファイルを作成したプログラムです。

Producer 実際に何らかのファイル形式から PDF へと変換したプログラムです。

CreationDate PDF の作成日時です。

Tagged アクセシビリティの向上のためにタグ付けされているかどうかです。

Pages ページ数です。

Encrypted 暗号化されているかどうかです。暗号化されているときは暗号化の内訳が表示されます (2.5.3 節)。

Page size 用紙のサイズです [pt]。

File size ファイルの容量です [byte]。

Optimized モニター用に最適化されているかどうかです。

PDF version PDF のバージョンです。

Xpdf 付属のユーティリティには PDF を PostScript ファイルに変換する `pdftops` があります。コンソールから `pdftops file.pdf` とするだけで `file.ps` が作成されます。

PDF ファイルからテキストのみを抽出したいときは `pdftotext` が使えます。`pdftotext file.pdf` とするだけで `file.txt` が作成されます*²⁴。

Xpdf 付属のユーティリティ以外にも Sid Steward 氏による PDFtk*²⁵も有用です。Windows であれば GUI 上から PDFtk を操作可能な GUI for PDFTK*²⁶もあります。PDFtk の使い方を解説した本の日本語訳も出版されています [28]。また、Hans Hagen 氏らによる ConTeXt というツール群に含まれている `texexec` を使うと複数の

*²⁴ エンコーディングの問題で正常に全ての文字を抽出できるとは限りません。

*²⁵ <http://www.accesspdf.com/pdftk/>

*²⁶ <http://www.clubic.com/telecharger-fiche13148-gui-for-pdftk.html>

PDF を操作する事ができます。

▼ 2.5.4 DVI を PostScript に——dvips

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事ができます。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 形式のファイルは多くの環境において Ghostscript と呼ばれるプログラムを使う事により、コンピュータ上で閲覧したり、プリンターで印刷する事ができます。

Tomas Rokicki 氏が開発した（そして Karl Berry 氏が Kpathsearch に対応させた）dvips を使うと DVI ファイルを PostScript ファイルに変換できます。dvips 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かも知れませんが、Red Hat の場合は pdvips という名前になっています。使い方はコンソールなどから次のようにするだけです。

```
$ dvips file.dvi
```

設定によっては直接プリンタにファイルが送信される場合があります。このような場合は次のように `-o` オプションを付けます。

```
$ dvips -o file.ps file.dvi
```

拡張子 `.dvi` は省略しても構いません。この dvips 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。
- T (横幅), (高さ) 用紙の大きさを単位付きで直接指定します。'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}}
```

としても同じ事になります。

- A 奇数ページだけ出力します。
- B 偶数ページだけ出力します。

- 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 \LaTeX 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 \LaTeX 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 \LaTeX のページ番号に依存します。11,21-35 のようにコンマで複数ページ指定する事もできます。
- r 印刷するページの順序を逆順にします。
- 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。Windows の方は常に config.dl を読み込むために


```
$ dvips -P dl -o filename.ps filename.dvi
```

 などとするのが良いでしょう。



複数ページからなる DVI ファイルの特定のページだけを EPS 形式にするならば

```
$ dvipsk -E -Pd1 -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。

▶ **問題 2.5** \LaTeX ファイルをタイプセットした $\langle file \rangle$.dvi は dvips で $\langle file \rangle$.ps へと変換する事ができますが、この PostScript ファイルを編集する事ができれば便利です。これには Angus Duggan 氏の psutils というツール群が役立ちます。ページの再配置や面付け作業などもこの psutils で行うと良いでしょう。実際にどのような機能があるのか、プログラムを実行し、その結果を吟味してください。

▼ 2.5.5 \TeX から HTML へ—— \TeX 4ht

\LaTeX の原稿ファイルを HTML に変換する事もできます。近年では自分が作成した文書を WWW 上で公開する事が頻繁にあります。例えば、教職員であれば数式を大量に含むような講義資料をネットワーク上に公開するときには、HTML で出力すると重宝すると思います。Unix 系 OS であれば \LaTeX 2HTML, TtH などが有名です。

本書では Eitan Gurari 氏が開発している \TeX 4ht の使い方について解説します。 \LaTeX 2HTML は \TeX 4ht に比べれば日本語情報がウェブ上にありますので、そちらを参照してください。

HTML への変換に必要なプログラムは NTT が開発した \TeX である $\mathit{j}\TeX$, 画像編集プログラムの ImageMagick, \TeX 4ht 本体です。ImageMagick は Windows でも

バイナリが用意されていますし、Unix 系 OS ならばパッケージに含まれている事が多いようです。

使用方法は `tex4ht` パッケージを原稿のプリアンブルに次のように記述します。

```
\usepackage[html,charset=Shift_JIS,png]{tex4ht}
```

次にコンソールから `ht jlatex file` とすれば `(file).html` と数式や画像などの PNG ファイルが出来上がります*27。 `ht` を実行するときのコマンドラインオプションで知っておくと便利なものに次のようなものがあります。

`-cleanup` HTML ファイルを生成後に中途ファイルを削除します。
`-output-name=<名前>` 出力ファイル名を<名前>に指定します。
`-output-dir=<ディレクトリ>` 出力するディレクトリを<ディレクトリ>に指定します。すでに存在するディレクトリでないといけないかもしれません。

▷ **例題 2.6** `ht` コマンドがどのような働きをしているのか、以下のようなコマンド列を実行して、その様子を確認してください。

```
$ jlatex (file).tex
$ jlatex (file).tex
$ jlatex (file).tex
$ tex4ht (file).tex ((file).dvi から(file).html の生成)
$ t4ht (file).tex ((file).css と画像の生成)
```

`ht` コマンドは基本的には上記の処理を連続して実行するプログラムです。一連の動作を示すと図 2.2 となります。

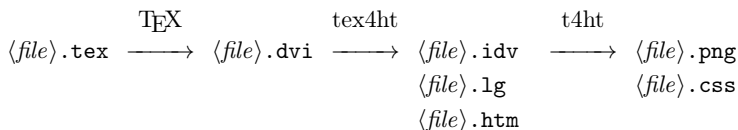


図 2.2 T_EX4ht の動作の概要

別の変換方法としてソースファイルで `tex4ht` を読み込まずにコンソールから直接

```
$ htlatex filename "html,charset=Shift_JIS,png"
```

*27 数式の画像化に関しては ImageMagick 等の外部プログラムを必要としますので、それらの設定が適切に行われていないと画像の表示はうまくいきません。

としても変換できます。

日本語の文書クラスを使うときは `j-article` `j-report`, `j-book` を使うようにします。そのための下準備として `$texmf/tex/generic/tex4ht/` などのディレクトリに移動し、

```
$ cp article.cls j-article.cls
$ cp report.cls j-report.cls
$ cp book.cls j-book.cls
```

として `j-classes` 用に設定ファイルを複製します。そうすると次のように日本語クラスファイルを使う事ができます。

```
\documentclass[11pt]{j-report}
```

`tex4ht` を読み込むときのオプションとして以下のものを追加すると良いでしょう。

html `tex4ht` のオプションで一番最初に指定するのは出力するファイルの形式です。

HTML (`html`) や XHTML (`xhtml`) などの形式を指定します。

charset=〈エンコーディング〉 文字コードを〈エンコーディング〉で指定します。

‘Shift_JIS’ と指定しても一部の半角英字が正しく表示されません。

fonts+ 標準のフォント設定では少し寂しいものがあるときは直接フォントをウェブブラウザに指定します。該当フォントがない場合は代替フォントに置き換わります。

fn-in 標準では脚注や傍注がおそらく別ページに出力されますが、このオプションを使うとページ最下部に出力されるようになります。

png 標準での画像出力形式は GIF (`gif`) になっていると思いますが、GIF の場合はバグなのかどうか分かりませんが、不正な GIF が生成される事もあるので PNG にしたほうが良いでしょう。ただし閲覧者のウェブブラウザが PNG 形式の画像を表示できるかどうかは分からないので注意が必要です。ここ最近のブラウザならば PNG は表示できると思われます。他に JPEG (`jpg`) も指定できます。

imgdir:〈ディレクトリ〉/ 標準では画像は HTML ファイルと同じディレクトリに出力されるのでこのオプションを指定して〈ディレクトリ〉を指定します。最後のスラッシュは必須と思われます。

pic-m 数式を画像化するオプション。 `tex4ht` は画像化しなくても良いと思われる部分は画像にしません。われわれ日本語圏の人間にはそれでは都合が悪い事があるので、苦渋の選択で全ての数式を画像化します。もっと強力に画像化するときは `pic-m+` を使います。 `equation` などの数式環境全体を画像化するならば `pic-equation`, `pic-equarray`, `pic-matrix`, `pic-array`, `pic-align` などを使

います。

pic-eqnarray 併用するパッケージによっては eqnarray 環境が正しく認識されないためかタイプセットできないので eqnarray 環境だけは画像化するように設定したほうが無難かもしれません。

(数字) 1 から 4 までの数字を指定して、出力 HTML ファイルのページを階層ごとに区切ります。L^AT_EX での見出しの階層に従って区切られます。

section+ 通常は目次からリンクを辿りますがこのオプションが指定されている場合は見出しから目次に戻る事ができます。

next DVI 形式や PDF 形式のファイルは連続的にページが続いています。しかし HTML 形式でファイルを出力すると不連続になりますので、連続的に次のページへ進むためのリンクを作成します。

htm 出力 HTML 形式のファイル名を(8 文字).htm とします。他の OS との互換性を考慮するならば必要かもしれません。減多にないと思いますが、例えば出力された HTML ファイルを ISO9660 フォーマットの CD-R に書き込むときなどに使えます。

以上のような設定をプリアンブルに次のようにすると良いでしょう。

```
\usepackage[html,charset=Shift_JIS,fonts+,fn-in,png,imgdir:images/,pic-m,pic-eqnarray,info]{tex4ht}
```

tex4ht は一番最後に読み込むようにするのが基本です。

他にも使用するパッケージがあるならば tex4ht の前に読み込んでください。

```
\usepackage[dvips]{graphicx,color}
\usepackage{url}
```

hyperref とは競合するようですから、次のようにすると良いでしょう。

```
\usepackage{その他のパッケージ}
\usepackage[オプション]{tex4ht}
\usepackage[tex4ht]{hyperref}
```

標準では以下の文字が日本語環境だと化けます。

```
\S \P \pounds \OE \ae \AE \aa \AA \ss \l \L
\o \O \i \j ? ‘ ! ‘ \textvisiblespace \textless
\textgreater
```

文字コードの iso-8859-1 は画像にしなくても良い文字なのですが、文字化けのため日本語環境では表示可能ではありません。uhtlatex を使って Unicode フォントに置

き換え、欧文フォントも Unicode にすると表示できると思われます。

日本語処理でひとつ問題となるのは余計な部分に入る半角空白です。この半角空白は `tex4ht` が文字列の処理を基本的に行単位で行い、その行を段落タグ '`<P>`' で閉じてしまうからです。欧文の場合は適切な区切りで文字が改行されるのでこの方法でも良いのですが、和文の場合はこれではいけません。しかし、適当な解決策は私にも分かりません。

既存の画像を挿入したいときは `\Picture` 命令を使います。

```
\Picture[⟨代替文字⟩]{⟨画像ファイル名⟩}
```

ある範囲を画像化するときには `\Picture+` 命令と `\EndPicture` 命令で囲みます。

```
\Picture+{⟨出力ファイル名⟩}{⟨要素⟩}\EndPicture
```

例えば `hoge.jpg` というファイルが存在し、これを HTML ファイル中に貼り付けるときは次のようにします。

```
\Picture[hoge の画像です]{hoge.jpg}
```

`tabular` 環境などの表全体を画像化するならば次のようになります。

```
\usepackage[html,png]{tex4ht}
\Picture+[画像にした表です]{mytable.png}
\begin{tabular}{|c|c|c|}
\LaTeX\,2.09& \LaTeXe& \LaTeX3\
\end{tabular}
\EndPicture
```

HTML タグを直接出力ファイルに埋め込むには `\HCode` 命令を使います。HTML タグの強制改行や水平線を入れるならば次のような使い方もできます。

```
\HCode{<HR><BR><BR><BR><BR>}
```

`hoge` 環境を使用しており、その環境を丸ごと画像にしたいならば `document` 環境の中で、次のような設定をすると `hoge` 環境が画像化されます。

```
\ConfigureEnv{hoge}
  {\IgnorePar\EndP{Tg<div class="pic-hoge">\Picture*{}}
  {\EndPicture{Tg</div>}{}}
\Css{div.pic-hoge {スタイルシート}}
%\ConfigureEnv{hoge*}%これは適宜記述してください。
% {\IgnorePar\EndP{Tg<div class="pic-hoge-star">\Picture*{}}
% {\EndPicture{Tg</div>}{}}
%\Css{div.pic-hoge-star {スタイルシート}}
```

picture 環境などの L^AT_EX で標準の描画用の環境は自動的に画像として出力されます。



索引や参考文献なども作成している場合は ht や htlatex などでは対応しづらいため、自分でスクリプトやバッチファイルを作成します。以下のようなシェルスクリプト tex4html を作成し PATH の通っている場所にコピーすると良いでしょう。

```
#!/bin/sh
jlatex $1
jbibtex $1
jlatex $1
jtex "\def\filename{{\$1}{idx}{4dx}{ind}} \input idxmake.4ht"
jmakeindex -o $1.ind $1.4dx
jlatex $1
jlatex $1
tex4ht $1
t4ht $1 $2
```

このファイルを `tex4html test` とすれば参考文献一覧のページや索引ありの HTML に変換できるはずですが、Windows の方は '\$1' を '%1' に書き換えてください。索引を作成していないときは途中で `test.idx` がいないためにエラーになるときがありますが構わず改行を押せば大丈夫でしょう。tex4ht では (NTT J^EX を用いる必要がある理由により) mendex を使う事はできませんので jmakeindex を使う事になります。さらに索引の作成方法は少し特殊で

```
$ jtex "\def\filename{{file}{idx}{4dx}{ind}} \input idxmake.4ht"
$ jmakeindex -o file.ind file.4dx
```

のように実行しないと索引が出力されません。mendex をいつも使っていて辞書ファイルなどで「読み」を別ファイルに保存している方は注意が必要です。

tex4html の二つ目の引数に t4ht に渡す引数を書く事もできます。これは

```
$ tex4html test "-p"
```

のような指定をして画像を生成しないように挙動を変える事もできます。

ハイパーリンクを作成するには `\Link` 命令も使えますが、個人的には `hyperref` か `url` パッケージを用いたほうが汎用性が高い記述になるでしょう。

```
\usepackage[html,charset=Shift_JIS]{tex4ht}
\usepackage[tex4th]{hyperref}
\href{http://www.google.co.jp}{Google}は検索エンジンです。
Googleを参照するにはウェブブラウザのアドレス欄に
\begin{quote}
\url{http://www.google.co.jp}
\end{quote}
と打ち込んで移動してください。
```

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、その理論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が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いません。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。日本語は漢字や仮名文字がありますが最小単位は**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、様々な事を半自動的に行ってくれます。この章では \LaTeX におけるそれらのルールについて解説します。

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次の三つを書き込みます。

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

\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。

例として入力以下に示すようなものと仮定します。

```
\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}
```

大体の出力は以下のようになります。

はじめての L^AT_EX

未来 太郎

2004 年 3 月 30 日

L^AT_EX を使うのは...

3

▷ **例題 3.1** 著者が複数人いるときは `\and` を使って区切ります。所属などがあるときは `\thanks` で欄外に出力します。以下のようなソースを実際に自分で出力してみてください。またその結果も吟味してください。

```
\author{夏目漱石\thanks{○○研究所 ○○事業部} \and
福澤諭吉\thanks{△△株式会社 △△研究所}\and
芥川龍之介\thanks{□□大学 □□学部 □□学科}}
```

▷ **例題 3.2** もう少し派手に著者名を紹介するとき、例えば「氏名」、「所属」、「連絡先」の三つを記述したいときは次のようにします。

```
\author{日本太郎\\ ○○大学 □□学科\\ name@server.ac.jp}
```

このソースを実際に出力しその結果を吟味してください。例題 3.1 の `\thanks` 命令と `\and` 命令を使います。

```
\author{夏目漱石\thanks{○○大学 ○○学科 name@server.ac.jp}}
```

もしくは改行で区切る方法のどちらも試してください。

▶ **問題 3.3** 以下のソースを出力してその結果を吟味してください。

```
\author{夏目漱石 \\ ○○研究所 \\ ○○事業部 \and
福澤諭吉 \\ △△株式会社 \\ △△究所\and
芥川龍之介\\ □□大学 □□学部 \\ □□学科}
```

この場合は横に著者が並びますが '`\and`' を取り除いて '`\\`' に書き換えると出力はどう変わるでしょうか。

▷ **例題 3.4** クラスファイルによる表題の体裁では不都合があるとき、その場しのぎ的には次のように調整します。

```

\begin{center}
{\LARGE \textbf{はじめての\LaTeX }}\\[2em]
{\large 未来 太郎}\[1em]
{2004年 3月 30日}\[1em]
\end{center}

```

本来はクラスファイル中の `\maketitle` 周辺のコマンドを適切に調整するのが望ましいでしょう。

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。

表 3.2 L^AT_EX での見出しの定義

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

`\section` などの見出し命令を使って見出しを作成します。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はなく、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、 \LaTeX 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。

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

▼ 3.3.2 見出しの深さ

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**に分ける事ができます。さらにその項目を小項目で分ける事もできるわけです。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事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。

\LaTeX 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のコマンドを用意しています。ただし (j)article など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。これは数十ページ以上の文書に存在する事が望まれます。目次といっても \LaTeX には次の三つの命令が用意されています。

<code>\tableofcontents</code>	(目次 [Contents] を出力するための命令)
<code>\listoffigures</code>	(図目次 [List of Figures] を出力するための命令)
<code>\listoftables</code>	(表目次 [List of Tables] を出力するための命令)

目次を表示するためには、それぞれ出力したい場所に命令を書きます。注意すべき事として、**目次を作成するためには最低2回のタイプセットを行います。**

その理由を考えるために、以下のようなファイル `mokuji.tex` を作ります。

```
\documentclass{jarticle}
\begin{document}
\tableofcontents%目次を出力する命令
\section{序論}
\subsection{研究背景}
\section{手法}
\subsection{実験環境}
\end{document}
```

次にこのファイル `mokuji.tex` をタイプセットすると出力ファイル `mokuji.dvi` にはただ「目次」と出力されるだけで、実際の目次が出力されていない事に注目してください。コンソールには次のようなメッセージが表示されます。

```
「 No file mokuji.aux.
  No file mokuji.toc.
  [1] (./mokuji.aux) 」
```

No File なんとか というのは(なんとか)というファイルが存在しない、足りないというメッセージです。ですが、同じディレクトリ(フォルダ)には `mokuji.aux` も `mokuji.toc` も存在するようです。どうやらタイプセットする前には存在せず、タイプセット後にこの二つのファイルは作成されたようです。この二つのファイルを覗いてみましょう。まずは中途ファイル `mokuji.aux` には次のように、ちょっと分かりづらい記述があります。

```
\relax
\@writefile{toc}{\contentsline{section}{\numberline{1}序論}{1}}
\@writefile{toc}{\contentsline{subsection}{\numberline{1.1}研究背景}{1}}
\@writefile{toc}{\contentsline{section}{\numberline{2}手法}{1}}
\@writefile{toc}{\contentsline{subsection}{\numberline{2.1}実験環境}{1}}
```

要するに `mokuji.toc` に見出し関係の記述をどうすべきかを書いているようです。さらに目次用の中途ファイル `mokuji.toc` には先ほどの `mokuji.aux` とほとんど同じ記述があります。


```
\contentsline{section}{\numberline{1}序論}{1}
\contentsline{subsection}{\numberline{1.1}研究背景}{1}
\contentsline{section}{\numberline{2}手法}{1}
\contentsline{subsection}{\numberline{2.1}実験環境}{1}
```

これを踏まえてもう 1 度 `mokuji.tex` をタイプセットします。するとコンソールには次のようなメッセージが表示されます。

```
(d:/usr/local/share/texmf/ptex/platex/base/jarticle.cls
Document Class: jarticle 2004/02/25
) (./mokuji.aux) (./mokuji.toc) [1] (./mokuji.aux)
```

どうやら `mokuji.aux` も `mokuji.toc` も存在し、それらを適切に処理してくれたようです。出力ファイル `mokuji.dvi` の先頭には始め出力されなかった「目次」がある事でしょう。

目次

1	序論	1
1.1	研究背景.....	1
1	手法	1
1.1	実験環境.....	1

▼ 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などで章 (`\chapter`) まで出力したいならば次のようにします。

```
\seccounter{tocdepth}{0}
```

(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。小節 (`\subsection`) までに番号を付けるようにするには次のようにします。これは目次側にも影響します。

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

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 ‘*’ を付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。例として次の記述をしてから概要の文章を書きます。

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

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。用途は異なりますが奥村晴彦氏の jsbook には abstract 環境が定義されています。これは各章の始めに「この章について」のようなまえがきを書くときに使われます。

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事ができます。具体的には次のようにすると目次にも概要を番号なしで書き出します。

```
\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}
```

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 \LaTeX は半自動で行います。使い方は 1 行空けて入力すれば良いだけです。

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。

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。

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。

このように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事ができ、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できます。‘`\parindent = 3zw`’のようにすると約全角 3 文字分の字下げを段落の始めで行う事ができます。

*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。

▼ 3.6.1 行頭の字下げ

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。

<code>\indent</code>	(可能ならば字下げをします)
<code>\noindent</code>	(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。

```
\usepackage{indentfirst}
\noindent 私は\indent 大学生ですか      私は大学生ですから、そうなります。
ら、そうなります。 \par                  そうなりました。
\noindent そうなりました。
```

▶ **問題 3.5** 行頭の字下げをせずに段落と段落に空きを入れて段落の終わりと段落の始まりを示すという事を \LaTeX で行うためには、段落と段落の空きを調節する `\parskip` という可変の長さ変数を調節します。

```
\parindent = Opt % 字下げ
\parskip = 10pt plus Opt minus Opt
```

実際にこのような設定にすれば分かりますが、ありとあらゆる部分に空気を挿入しますので、その出力結果を吟味してください。

▼ 3.6.2 段落の字下げ——`indent`

文を引用している場合はそれが引用である事を明確にするために、段落全体を字下げする習慣があります。これには `quote` 環境や `quotation` 環境が使えます。ただしこの場合は自分で字下げ幅を設定できません。簡単に段落の字下げを調整するには `indent*2` パッケージの `indentation` 環境を使います。

<pre>\begin{indentation}{左側の字下げ}{右側の字下げ} <文章内容> \end{indentation}</pre>

*2 CTAN: macros/latex209/contrib/misc/indent.sty

`indentation` 環境の一つ目の必須引数には**左側**の字下げ、二つ目には**右側**の字下げを指定します。

ここは普通の文章領域です。不必要に字下げを調整するのは好ましいことではありません。

```
\begin{indentation}{3zw}{3zw}
左側の字下げは全角 3 文字分、右側の
字下げも全角 3 文字分ありますか？
\end{indentation}
\begin{indentation}{0zw}{5zw}
左側の字下げはなし、右側の字下げ
は全角 5 文字分ありますか？
\end{indentation}
ここも普通の文章領域です。
```

ここは普通の文章領域です。不必要に字下げを調整するのは好ましいことではありません。

左側の字下げは全角 3 文字分、右側の字下げも全角 3 文字分ありますか？

左側の字下げはなし、右側の字下げは全角 5 文字分ありますか？
ここも普通の文章領域です。

▼ 3.6.3 ダブルスペース

ダブルスペースといって**行送り**を倍にするという事を迫られる場合があります。これには Geoffrey Tobin 氏による `setspace` パッケージを使う事が考えられます*3。

<code>\singlespacing</code>	(通常通りの行送りに設定する)
<code>\onehalfspacing</code>	(通常の 1.5 倍の行送りにする)
<code>\doublespacing</code>	(通常の 2 倍の行送りにする)
<code>\begin{spacing}{<数値>} <文章> \end{spacing}</code>	

<数値>を指定して行送りを変更できる `spacing` 環境も用意されています。

<code>\usepackage{setspace}</code>	
<code>\singlespacing</code>	ここは通常の
ここは通常の <code>\par</code> 行間 <code>\par</code>	行間
<code>\doublespacing</code>	ここは通常の
ここは通常の <code>\par</code> 2 倍の行間 <code>\par</code>	
<code>\begin{spacing}{.8}</code>	2 倍の行間
ここは通常の <code>\par</code> 0.8 倍の行間 <code>\par</code>	ここは通常の
<code>\end{spacing}</code>	0.8 倍の行間

*3 他にも `doublespace` パッケージを使う方法や `\baselinestretch` 命令を再定義する方法もあります。

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）に数値を代入する時に ‘\parindent=0pt ’ という記述がありました。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れ

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）	実際の長さ
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	┌┐

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*4。

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 (×) 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。3mkg (×) は正しくは 3g となります。

*4 この他にも 10²⁴ 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、頻繁に用いられるだろう修飾子だけを掲載しました。

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**。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

句読点は、文を区切るために文間に挿入する記号です。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も違います。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。

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。

横書きのときは句読点は**半角**のコンマとピリオドを使うのがベストだと思います。その場合は句読点の後には半角のスペースを入れてください。横書きで全角の句読点を使うときは

今日のお弁当のふたには“I wish I could.”という文句が書かれていた。

のように二つの文書に異なる句読点が混在する事になります。なるべく句読点は統一したほうが気持ちが良いでしょう。ただし、

今日のお弁当のふたには「I wish I could。」という文句が書かれていた。

などとするのはとっても気持ち悪いのでやめたほうが無難です。横書きのときは文章の内容と相談して句読点を決めてください。

句読点が少し気持ち悪く見える例です。

地球は、青かった。 10\,m, 高かった。 地球は、青かった。 10m, 高かった。

どのようにするのが良いかはご自分で判断していただければ良いのですが、本書でも一応の見解を出しておきます。まずは欧文だけの場合を考えます。欧文のみの文書では以下の規則に従うだけで良いでしょう。

- 句読点は半角のコンマ「,」とピリオド「.」を使う。
- 単語の引用はシングルクォート「'」, 一文の引用はダブルクォート「“”」を使う。

和文のみの場合は

- 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ォート「『』」を使う。

となるでしょう。ですが和文と欧文が混在する文書ではどのように取り決めたら良いのでしょうか。本来ならばまず句読点のつけ方も引用の仕方も一つの文書の中で統一す

るのが規則です。そのため和文と欧文が混在する場合は欧文の規則に従うという事です。ですが、そうすると別の問題が出てきます。日本語組版で「句読点は1文字分の幅を割り当てる」という規則に反する事があります。これには全角のコンマ「、」とピリオド「。」を使うようにすると我慢できます。

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。

```
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*5。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`\footnote`
{Fourier Translation}は通常理工系の
大学ならば必修で\dots と思われる。

ラプラス変換やフーリエ変換⁴は通常理工系の
大学ならば必修で...と思われる。

⁴ Fourier Translation

3.10 文字の強調

ビジネス文書等では重要な文に下線 (underline) を引いて強調 (emphasis) を表しているものがあります。論文や書籍の場合は**欧文をイタリック体**、**和文はゴシック体**にします。L^AT_EX では文字列の強調のために `\emph` 命令が使えます。

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。

場合によっては `\em` コマンドが使えます。これは**宣言型コマンド** (4.2.2 節) と呼

*5 このように注釈が文章の頁の下端に出力されます。

ばれるもので、広範囲な論理強調に使う事ができます。ただしこのコマンドを使うならば**イタリック補正**を挿入する場合があります。イタリック補正とはイタリック体の文字の直後にローマン体などの別のフォントが来た場合に挿入すべき空白の事です。L^AT_EX が用意している `\emph` や `\textit` などはこのイタリック補正が適切に挿入されるようになっています。宣言型の強調コマンド `\em` を使う場合は自分で挿入しますから `\` 命令を使う事になります。

```
\emph{Future University-Hakodate}
is interesting.\par
{\em FUN\} is funky?\par
{\em FUN} is not funky?
```

Future University-Hakodate is interesting.
FUN is funky?
FUN is not funky?

3.11 そのまま出力できない記号

L^AT_EX ではいくつかの半角の記号を直接出力する事ができません。これは予約文字と呼ばれるもので、`\` に似た命令の一種だと思ってください。出力できない記号は

```
\ { } $ & # ^ _ ~ % < > |
```

の 13 個であり、それらの記号を出力するには 2.1.3 節 を参照してください。

▼ 3.11.1 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、それらを出力するには表 3.7 の命令を uses。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 uses。`'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 uses。

```
J\ "org {mu\ss} ein Gel\ "ande f\ "ur
  seine Fabrik erwerben.      Jörg muß ein Gelände für seine Fabrik er-
                               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

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	ĵ \j	¡ !‘	þ \th *	> \guilsinglright *
œ \oe	ß \ss	¿ ?‘	Þ \TH *	„ \quotedblbase *
ƒ \OE	Š \SS	ð \dh *		, \quotesinglbase *
ł \l	§ \S	Đ \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}	ǎ \H{a}	ạ \b{a}	ọ \k{o}	ñ \~{n}	
á \. {a}	ô \^ {o}	ç \c{c}	ĩ \u{ı}	ôo \t{oo}	

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。

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。

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、ある行のどこからでも ‘%’ があるとそれ以降を**コメント**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}
```

ここは出力されますが % ここはされない。

```
\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？

ここは出力されますが出力されませんか？

3.14 べた書き

テキストをそのまま出力するときがあると思います。例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**べた書き** (verbatim) が可能です。短い文字列の場合は `\verb` 命令を使います。

```
\verb+(文字列)+
```

複数行になるときは `verbatim` 環境を使います。

```
\begin{verbatim}
<ここにべた書きしたい複数行の文字列を挿入します>
\end{verbatim}
```

`\verb|#include<stdio.h>|` はプリプロ

セッサとよばれ...

```
\begin{verbatim}
```

```
int main( void ){
```

```
    int foo = 0x7E;
```

```
    printf("%c\n", foo);
```

```
}
```

```
\end{verbatim}
```

`#include<stdio.h>` はプリプロセッサとよばれ...

```
int main( void ){
```

```
    int foo = 0x7E;
```

```
    printf("%c\n", foo);
```

```
}
```

`\verb` 命令や `verbatim` 環境にはアスタリスクを付ける事ができます。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事になっています。

```
\verb|2323 ^_^;|, \verb9|()|9.
```

```
\verb*|1 3 5|, \verb*9ok? ok?9.
```

```
\begin{verbatim*}
```

```
int main( void ){
```

```
    printf("Hello, World!\n");
```

```
}
```

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

```
2323 ^_^; |()|. 1_3_5, ok?_ok?.
```

```
int_main(void){
```

```
    printf("Hello, World!\n");
```

```
}
```

▼ 3.14.1 特殊なべた書きその 1——alltt

Johannes Braams 氏が作成したべた書きのもう少し気の利いた環境が使える `alltt` パッケージがあります。このパッケージでは `\`, `{`, `}` が通常通りの命令として使え

る `alltt` 環境を提供しています。

```
\usepackage{alltt}
\begin{alltt}
{\em This is emphasized}.
Dollar $$ is not allowed, but
{\LaTeX's} math environment
\(\int f(x)dx\) is OK!!
\end{alltt}
```

This is emphasized.
Dollar $\$x\$$ is not allowed, but
L^AT_EX's math environment
 $\int f(x)dx$ is OK!!

▼ 3.14.2 特殊なべた書きその2——`cmmt`

`\texttt` や `\ttfamily` で書体をタイプライタファミリーに変更したとしても、波括弧やバックスラッシュはそのままでは出力できませんし、書体がローマンファミリーになります。 `\verb` 命令を使えば良いではないかと思われそうですが、引数の中では使えない、区切りに使っている記号は使えないなどの不都合があります。この場合は Mark Wooding 氏による `cmmt` を使うと良いでしょう。

```
\textmmtt{<文字列>}
\mmtfamily (宣言型のコマンド)
\mmt{<文字列>}
```

`\mmt` 命令中で L^AT_EX の予約文字を出力するには次のようにします。

```
\\ { \{ } \} _ \_ ^ \^ $ \$ % \%
& \& # \# ~ \~ " \" ' ' _ \_ | \|
```

```
\begin{minipage}{.9\linewidth}
\texttt{\{foo\} \& bar}\par
\textmmt{\{foo\} \& |bar|}%
\footnote{\mmt{\ttfamily\{any\}}}{\ttfamily{any}}
\end{minipage}
```

{foo} & bar
{foo} & |bar|^a

^a \ttfamily{any}

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。L^AT_EX 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L^AT_EX 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事になります。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 quo-
tation 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。

どちらか一方に統一するのが作成者側にも読者にも混乱は少ないでしょう。L^AT_EXで引用というものをもう少し効率良く行うには次のように定義します。

```
\newcommand{\qu}[1]{‘#1’}% 単語の欧文引用
\newcommand{\qq}[1]{‘‘#1’’}% 1文の欧文引用
\newcommand{\yo}[1]{「#1」}% 単語の和文引用
\newcommand{\yy}[1]{『#1』}% 1文の和文引用
```

こうしておけば、後から引用符を統一できます。上記の\yoと\yyを次のように変更するだけで文章中の引用符を一括して変更できるわけです。

```
\newcommand{\yo}[1]{‘#1’}% 単語の和文引用
\newcommand{\yy}[1]{‘‘#1’’}% 1文の和文引用
```

それは渡辺らによれば\yo{デカルトの名言に{\qu{I think, therefore I am.}}がある}という調査結果が存在する。

それは渡辺らによれば「デカルトの名言に‘I think, therefore I am.’がある」という調査結果が存在する。

▼ 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{〈欧文の文献名〉}
『〈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。



このような他の文書の引用には新たに欧文用の引用命令 `\yousyo` や、和文用の `\wasyo` などを作るとあとで統一したいときには便利でしょう。

```
\newcommand{\yousyo}[1]{\emph{#1}}%欧文
\newcommand{\wasyo}[1]{『#1』}%和文
```

▼ 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事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 young---a little
dog---I could read about 100--200
books in a day. This is a just
fairy-tale.’’

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

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。

表 3.9 ダッシュなど

記号の種類	出力	入力・命令	説明
en-dash	–	--	ハイフンを二つ
em-dash	—	---	ハイフンを三つ
全角ダーシ	—	—	全角のダッシュ
倍角ダーシ	——	\---	'\'と全角ダーシ二つ
ハイフン	-	-	そのまま
マイナス	-	\$\$-	数式中でハイフン一つ

`{\TeX}`の`\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 ハイフネーション

行末に長い単語がある場合にその単語のハイフネーションにあった改行位置を見て `LaTeX` は改行を行います。ユーザが明示的に改行位置を指定する事もできます。

```
\hyphenation{⟨複数の語句⟩}
\-(位置指定)
```

`\-` 命令は文章中で直接 '`Ghost\script`' のように使います。 `\hyphenation` の場合はプリアンブルなどに次のような記述をしておきます。

```
\hyphenation{Ghost-script Post-Script}
```

```
George Johnson and George Brahms
always say
supercalifragilisticexpialidocious,
they always say super\cali\
fragilistic\expiali\docious, and
Robert sometimes say \mbox
{supercalifragilisticexpialidocious}.
```

```
George Johnson and George Brahms
always say supercalifragilisticexpialido-
cious, they always say supercali fragilistic-
expialidocious, and Robert sometimes say
supercalifragilisticexpialidocious.
```

‘super 云々’の一つ目、二つ目では改行の位置が違います。 `\mbox` 命令を使うと改

行を許しません。

▼ 3.15.4 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事が可能ですが、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。できる事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事が多いです。

```
\\*[(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事はできません。アスタリスクを付けると改行直後にページを改める事を禁止します。\\newline は ‘\\’ とほぼ同時の命令です。\\par は改行ではなく**改段落**、すなわち段の始まりを示します。その直後の文字列は \\parindent の値に応じて字下げされます。

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生まれ \\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生まれ
字下げされる場合があります。

3.16 空白の扱い

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。

▼ 3.16.1 文章の中の空白

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

ここでは‘f1’や‘ff1’などがその例です。

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**とか**単語間スペース**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.	I want to be a Mr. Right and go to N.Y. I wish I could.
--	--

‘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せんが、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 $\backslash_$ 命令を、文間空白には $\backslash_$ 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**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 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ が最適な空白の量を調節してくれているので、普段は気にする事はないでしょう。

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 L^AT_EX が最適な空白の量を調節してくれているので、普段は気にする事はないでしょう。

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事になります。

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。 `_` 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。 `\@` 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。 `\par` 命令で明示的に段落の終了を告げる事ができる。

▼ 3.16.2 その他注意すること

それらが並んでいる事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、略語などはひとまとめにします。これにはチルダ `~` を使います。

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

Mr. Sato read page 10 and looked at fig. 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 fine greeting and I
always say ‘Hello.’” \par
“\, ‘Hello’ is a fine greeting and
I always say ‘Hello.’\,”

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

▼ 3.16.3 和文と欧文のあいだの空白

日本語 T_EX 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 4 分 1 の空白の事です。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ります。例では `\mbox` で四分空きを無効にしています。

日本語と good のあいだには四分空きが
あると`\par` があると`\mbox{good}`です。

日本語と good のあいだには四分空きが
あると`good`です。

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がります。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の際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。

```
\LaTeX と日本語\TeX      \\  
\LaTeX\ と日本語 \TeX    \\  
{\LaTeX}と日本語{\TeX}  \\  
{\LaTeX} と日本語 {\TeX}\
```

```
LaTeX と日本語 TeX  
LaTeX と日本語 TeX  
LaTeX と日本語 TeX  
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。

いずれの方法においても欧文同士の空白に注意します。

```
\TeX and \LaTeX are famous.\  
{\TeX} and {\LaTeX} are popular.
```

```
TeXand LaTeXare famous.  
TeX and LaTeX are popular.
```

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。

このような空きの調整に対しては 8.15 節で解説している `xspace` パッケージが有用です。

3.17 行揃え

行揃えには三つの環境と三つの宣言型のコマンドを使う事ができます (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境の中で使う事ができます。中央揃えには `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。右揃えには `flushright` 環境です。文字列を右寄せにします。左揃えには `flushleft` 環境です。字下げを行わずに左に寄せます。

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>

ビジネス文書で大活躍するでしょう。

```
\begin{flushleft}
```

段落の下下げを行わずに \\ 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。

段落の下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。

```
\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。

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。

```
\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。

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 \\ 2004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}
```

このようにすると、以下のような出力となります。

3

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

3.18 箇条書き

箇条書きには次の三つの環境を用いる事ができます。

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) にする事が可能です。入れ子にでき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[#]’` とする事で先頭のラベルの記号を指定する事が可能です。

```

\begin{itemize}
  \item 入れ子にしたい.
  \item[*] 入れ子になる.
\begin{itemize}
  \item 入れ子です.
\end{itemize}
\end{itemize}

```

- 入れ子にしたい.
- * 入れ子になる.
 - 入れ子です.

```

\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.

▼ 3.18.1 番号付箇条書き環境の拡張——enumerate

番号付の箇条書き環境 `enumerate` の拡張を David Carlisle 氏が行ない、これを `enumerate` パッケージとして用いる事ができます。 `\Alph`, `\alph`, `\Roman`, `\roman`, `\arabic` という命令の変わりに、次の文字 (トークン) によってカウンタとすべき修飾子を決めます。

`A = \Alph` `a = \alph` `1 = \arabic` `I = \Roman` `i = \roman`

修飾子としたい文字は絶対に波括弧 `{ }` でグルーピングしません。逆に修飾子と区別が付かない文字列 (Exmample, A などなど) はグルーピングします。まずは、使用例を吟味してください。


```

\usepackage{okumacro,graphicx}
\usepackage{enumerate}
\begin{enumerate}[{例題} 1]
  \item\label{exe:a} 未来です.
  \begin{enumerate}[{問題}
    \ref{exe:a}-1]
    \item これはどうか.
    \item それはどうです.
  \end{enumerate}
  \item あれは云々.
  \item これは云々.
\end{enumerate}

```

例題 1 未来です.

問題 1-1 これはどうか.

問題 1-2 それはどうです.

例題 2 あれは云々.

例題 3 これは云々.



丸数字で番号付けをしたいとき等は `okumacro` パッケージの `\MARU` 命令を使います。このような特殊なケースは `enumerate` パッケージを使わずに直接 `\labelenumi` などのラベル部分を再定義した方が素直にできる場合もあります。

```

\begin{enumerate}
\renewcommand\labelenumi{\MARU{\arabic{enumi}}}
\item \TeX
\item \LaTeX\,2.09
\item \LaTeXe
\item \LaTeX\,3
\end{enumerate}

```

3.19 これまでの復習

▶ **問題 3.6** 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できます。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いません。

```

\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。

3.20 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。

世の中で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事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。

LaTeX はマークアップ型のシステムなのでユーザが直接書体変更用の命令を使う事は本来ならば必要のない事だと思われます。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。

はじめての L^AT_EX

自分の名前

2006年2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。段落ごと引用することは次のようになっている。

一つの段落の引用の場合は `quote` 環境を使い、行頭を字下げしないのが普通である。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

▼ 3.20.1 文字の大きさの変更

LaTeX においては比較的簡単に文字の大きさを変える事が可能ですが、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 (4.2.2 節を参照) を次のように使用します。

{<宣言> <文字の大きさを変えたい文字列>}

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>	やちよう
大きい	<code>\Large</code>	かちよう
かなり大きい	<code>\LARGE</code>	らいちよう
とても大きい	<code>\huge</code>	はくちよう
特大	<code>\Huge</code>	ひちよう

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。

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。

```
\newcommand\kyocho[1]{\Large#1}
\newcommand\Kyocho[1]{\LARGE#1}
だから\kyocho{ここは大事です}. それに
\Kyocho{ここはもっと大事}です.
```

だからここは大事です. それにこ
こはもっと大事です.

▼ 3.20.2 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.

文字の大きさや書体の種類を不必要に変更する事は, 逆に読者を混乱させるだけです. 自分はきちんと使い分ける事ができる, という方は表 3.13 の一覧から適切な書体を選んで, 美しい文書を作成してください.

ファミリーとシリーズとシェイプはそれぞれ組み合わせて使う事ができます. 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.

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

```
\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 Large Bold?} 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 p_LA_TE_X で和文の多書体を図る事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るので、徒に行わないほうが良いかもしれません。

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。

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？

▼ 3.20.3 基本書体の変更

フォントの大きさやファミリーなどを指定する命令は分かりました。しかし、文書に使われている書体そのものを変更するにはどうすれば良いのでしょうか。実は普段何気なく \LaTeX で文書処理をしているときに使われている欧文のフォントは Donald Knuth 氏がデザインした Computer Modern と呼ばれる**基本書体**になっています。この基本書体そのものを変更するには基本書体を変更する定義がなされたマクロを読み込むか、自分で指定する必要があります。数式に使われる数式書体にも基本的に Computer Modern が使われます。フォントについては奥村晴彦氏の文献 [32] などを参照してください。本書では取り扱いません。あえて言うならば Young Ryu 氏が作成した txfonts を使うのが手軽ではないかと思います。標準配布の Times 系のフォントを使うようにする mathpmtx や、Palatino 系のフォントを使うようにする mathpazo よりも pxfonts や txfonts という選択肢もあります (A.2 節 を参照)。

3.21 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事はほとんどありません。何度も修正と加筆を繰り返す、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。 \LaTeX 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あって、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つてはいけないという、行頭禁則処理の問題も \LaTeX (\pTeX 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 [61] 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事ができるので、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ない。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 [55, 57, 58] 上で注意すべき点として以下の項目などが挙げられます。

- 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。

▼ 3.21.1 構文チェック——`syntonly`

\LaTeX の原稿を書いていると、コマンド構文の記述ミスをしてしまうことがあります。括弧が足りないとか、大文字と小文字を書き間違えているなどです。Frank Mittelbach 氏と Rainer Schöpf 氏が作成した `syntonly` を使うと、自分の記述が間違っていないか、実際の出力をせずに構文のチェックだけを行うようにできます。`\syntaxonly` 命令をプリアンブルに記述します。`syntonly` パッケージを読み込んだだけでは何も起こりません。構文だけをチェックするので通常の DVI 出力を伴うタイプセットよりも処理が速いと思います。

```
\documentclass{jarticle}
\usepackage{syntonly}
\syntaxonly
\begin{document}
I like {\LaTeX}.%これはエラーになります。
\end{document}
```

3.22 クラスとパッケージ

\LaTeX はマークアップ方式を採用した言語なので書式と内容は分離されるのが一般的です。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きます。

\LaTeX 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、 \LaTeX 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 2 段組の記事を書こうと思えば次のように原稿中で宣言します。

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

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事を同時に宣言する事もできます。

```
\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* を付けるようにします。

▼ 3.22.1 標準的なクラス

\LaTeX や $\mathrm{p}\LaTeX$ 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事が多いようです。

あるクラスが何からインストールできるのかは、すでにでき上がっているクラスファイル *<classes>.cls* が存在すればそのファイルの先頭部分に次のような記述があります。

```
%% This is file 'class.cls',
%% generated with the docstrip utility.
%% The original source files were:
%% classes.dtx (with options: 'option')
```

オリジナルのソースファイルはオプションに *'option'* を指定してファイル名が *<classes>.dtx* です、と説明してくれていますので Unix 系 OS ならば

```
$ find /usr/local/share/texmf/ -name classes.dtx
```

などでそのファイルを検索できるでしょう。Windows や Macintosh はファイルの検索の機能があると思いますので、そちらを使えば良いでしょう。そのようにして検索した場所に `<classes>.dtx` や `<classes>.ins` が存在します。そのファイルがある場所に移動して `platex classes.dtx` とすればそのクラスの仕様書を見る事ができます。さらに `platex classes.ins` とするとそのクラスを導入する事ができます。このような操作をすると DocStrip というマクロが適切にファイルを処理します。例えばファイル `classes.ins` をこのように処理すると、`article.cls`, `report.cls`, `book.cls`, `bk10.clo`, `bk11.clo`, `bk12.clo`, `size10.clo`, `size11.clo`, `size12.clo` という九つのファイルが生成されます。

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。標準では以下の欧文用のクラスが使えます。

- article 小規模の記事を作成するためのクラス。 `classes.dtx` や L^AT_EX コンパニオンに詳しい仕様が書かれている。
- report 報告書を作成するためのクラス。同じく `classes.dtx` や L^AT_EX コンパニオンに詳しい仕様が書かれている。
- book 書籍を作成するためのクラス。同じく `classes.dtx` や L^AT_EX コンパニオンに詳しい仕様が書かれている。
- slides スライドを作成するためのクラス。 `slides.dtx` に詳しい仕様が書かれている。
- proc `article` をベースに議事録などを作成するためのクラス。 `proc.dtx` に詳しい仕様が書かれている。

以上の `article`, `report`, `book` の三つをまとめて `classes` と呼ぶ事があります。

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。
- tarticle 縦書きの小規模の日本語の記事を作成するためのクラス。
- treport 縦書きの日本語の報告書を作成するためのクラス。
- tbook 縦書きの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` と呼ぶ事があります。

▼ 3.22.2 クラスオプション

ドキュメントクラス（文書クラス）にはもう少し詳細な設定を行う事ができます。`\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 (`10pt`, `11pt`, `12pt`) 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 (`a4paper`, `a5paper`, `b5paper`, `letterpaper`) 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 (`landscape`) 用紙を横置きにします。標準は縦置きです。

印刷面 (`oneside`, `twoside`) 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 (`onecolumn`, `twocolumn`) 1 段組 (`onecolumn`) にするか 2 段組 (`twocolumn`) にするかを指定します。

表題 (`titlepage`, `notitlepage`) 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 (`fleqn`) 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 (`leqno`) 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 (`draft`, `final`) 執筆途中で印刷するときに `draft` オプションを付けると、文書幅の領域をはみ出してしまった箇所に印が付いてくれます。原稿が完成したら `final` に変更します。標準は `final` です。

左右起し (`openright`, `openany`) (`jreport` や `jbook` において章などの開始ページの指定をします。常に奇数ページで起こす (`openright`) か、どちらからでも起こす (`openany`) かを設定します。`jreport` の標準は `openany`。 `jbook` 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に定評があります*6。このクラス群を導入すると

*6 <http://oku.edu.mie-u.ac.jp/~okumura/jsclasses/>

- 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*7。以上の `jsarticle`, `jsbook`, `jspf` の三つをまとめて `jsclasses` と呼びます。

文字サイズ `<9pt, 10pt, 11pt, 12pt, 14pt, 17pt, 20pt, 21pt, 25pt, 30pt, 36pt, 43pt, 12Q, 14Q>`

用紙サイズ `<a4paper, a5paper, a6paper, b5paper, b4paper, a4j, a5j, b4j, b5j, a4var, b5var>`

言語の指定 `<english>` 欧文用の見出しの定義と行送りになります。

用紙サイズ情報 `<papersize>` 用紙サイズの情報をデバイスドライバに渡すようにします。

レポート作成 `<report>` レポート作成用に `\chapter` 命令を使う事ができます。 `jsbook` では左右起し等に関する設定が変わります。

▼ 3.22.3 標準で使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次のようにすると使用可能になります。

```
\usepackage[<オプション>]{<パッケージ>}
```

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*8。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` を検索する事はできません。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。

*7 (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。

*8 最近の TeX 環境では `texdoc` (パッケージ名) 等とすればマニュアルが表示されます。

\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事でしょう。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してされています。

\AMSLATEX 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。 \AMSTEX 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 \AMSLATEX で、 \LaTeX でのパッケージの名前は `amsmath` と言います。5.11 節で紹介しています。

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。詳しい情報は付録 B.5 を参照してください。

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ます。これはデバイス依存の機能で環境により出力が異なる事があります。6.5 節で紹介しています。

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こぼれたマクロ。

\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事ができるマクロ。
- `calc` \LaTeX での計算を楽にするマクロ。8.21.1 節で紹介しています。
- `dcolumn` 表や行列の環境で小数点などを揃えるためのマクロ。6.3.4 節で紹介しています。
- `delarray` 行列で括弧付けを容易にするためのマクロ。5.5.5 節で紹介しています。
- `hhline` 表や行列で複雑な罫線を簡単に引く事ができるマクロ。
- `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。6.3.6 節で紹介しています。
- `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。6.3.7 節で紹介しています。
- `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。5.9.3 節で紹介しています。
- `enumerate` `enumerate` 環境を拡張するためのマクロ。3.18.1 節で紹介しています。
- `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 (<code>\chapter</code>) や節 (<code>\section</code>) 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。3.6.1 節で紹介しています。
layout	現在の文書のページレイアウトを表示するマクロ。8.1.1 節で紹介しています。
multicol	多段組を実装するためのマクロ。8.7.1 節で紹介しています。
showkeys	<code>\label</code> , <code>\ref</code> , <code>\cite</code> などの相互参照のラベル名 (keys) を表示するためのマクロ。4.6.1 節で紹介しています。
theorem	定理型環境を簡単に宣言するためのマクロ。5.8.1 節で紹介しています。
varioref	相互参照をしやすくするためのマクロ。
verbatim	<code>verbatim</code> 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。8.15 節で紹介しています。

第 4 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 L^AT_EX 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。

4.1 マークアップ言語とは？

今は昔、文書に対して入れ子型の論理構造を与える事によって汎用性を持たせ、人間にも機械にも理解するのが簡単な文書の記述に関して研究がなされたそうです^{*1}。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 というものが出現したそうです。これは Tim Berners-Lee 氏らが開発し、今は W3C が管理しているページ記述言語です^{*2}。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。L^AT_EX 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。

マークアップ型の言語にはどのような利点があるのでしょうか。それをこの章では少し考えてみたいと思います。これを考えるには L^AT_EX の（命令・環境・宣言を含む）コマンドを観察する事から糸口が見えてくると思いますので、コマンドに関する話を中心に進めていきます。

4.2 記号とコマンド

L^AT_EX はコンピュータプログラムですから、人間の意志を理解するためには何か特別な命令を人間から受け付けなくてはなりません。

^{*1} HTML の先駆けである SGML が 1960 年代から軍事技術の一環として研究されてきました。

^{*2} 当時は学術論文用の記述言語という位置付けだったという話があります。

「もしかしてこの辺は太字にしたいのですか？」

などに対応する事はありません。皆無というわけではありませんが、基本的にこちらから命令するまで何もしない性格です。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**を持たせて使用します。ですから \LaTeX での記号の意味とコマンドの役割を知っておきましょう。

▼ 4.2.1 記号の分類

いきなり \TeX / \LaTeX の記号の分類を考えるよりも、普段使用している日本語や英語の記号と言語を考えましょう。私たちはコンピュータが理解するのが難しい言語を使っています。特に日本語はコンピュータが理解しづらいものとなっています。それでも私たち人間は文脈などを道筋に、それらの意味を読み取ります。

今日は1,000円を持って、佐藤さんの商店に塩を買いに行った。

という1文を考えると、人間はすぐに「1,000円」のコンマ‘,’を「お金の3桁目に入れる区切り」、最後の句点‘.’を「文の終わり」として理解できます。私たちが複雑な日常文を理解できる理由として、文の中に必ず**文法**が存在するという事が挙げられます。

人間はある程度曖昧な表現を見つけてもそれを柔軟に理解できますが、コンピュータはなにせ‘1’と‘0’しか分からない融通の利かない機械ですから、人間が日常的に用いている自然言語よりは正確な文法を持った、コンピュータが理解できる形式言語で記述してあげる必要があります。 \TeX / \LaTeX は生成文法を基礎とした形式言語しか理解できませんから、ユーザから入力される記号には全て明確な意味を持たせる必要があります。現在のバージョンの \TeX / \LaTeX では曖昧な意味を持たせて文章を執筆する事はできません。

\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理解してくれません。‘ $\$<\$$ ’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

$\backslash \{ \} \$ \& \# \^ _ \sim \%$

という 10 個の記号には特別な意味がある事を覚えてください。それぞれの記号の使い方はその時々説明しますので、今はざっと 10 個の記号が特別な意味を持ち、英文字と漢字、それに平仮名や片仮名だけが普通の単語を作るための文字として認識されるのだと解釈して頂いても構いません。



TeX/LaTeX は標準的には JIS X 0208 (JIS 第 1 水準・第 2 水準) の文字集合までしか扱う事ができませんから、機種依存文字や半角カナ文字は出力できません。これらを拡張する試みがいくつかあります。近年は齋藤修三郎氏による Adobe-Japan-1-6 までの文字集合を扱える OTF パッケージ等があります。機種依存がどの文字かを知るために、例えばウェブで検索するのも良いでしょう。

▼ 4.2.2 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 ‘j’ になってしまいます。これは一体どういう事でしょうか。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、‘j’ 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」という形式ではない事が分かります。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事もあります。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。‘\ ’ の事。

特殊記号 特別な意味を持つ記号。予約文字と呼ばれる事もあります。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.

の三つがある事を理解してください。本書では制御綴り（コントロールシーケンス）の事を**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事を**コマンド**と呼ぶ事が多いようです。引数を取る事があり、その引数の事を**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（グルーピングする）事もできます。引数をとる事は稀。よく宣言の事も**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` 等です。宣言型のコマンドは命令に比べると少ないので、本書でも断り書きとして宣言型コマンドと呼ぶ事が多いです。

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事。引数を取る事があります。例として `document` 環境などがあります。

▼ 4.2.3 コマンドの定義

LaTeX の原稿では新しい命令などの定義をする事ができます。

```
\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>を新規に定義する事ができます。

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

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。

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

続いて上記の記述に基づき本文中で `\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。さらにこの `\example` 命令に任意引

数があっても良い事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` 命令を使う事により実現できます。定義の中で引数は ' $\langle n \rangle$ ' 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` では任意引数を一つしか設けることができませんが、引数は合計 9 個まで使う事ができます。`\renewcommand` では一度定義した命令を再度定義する事ができます。

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{<環境名>}[<引数の個数>][<標準値>]{<始め>}{<終わり>}
\renewenvironment{<環境名>}[<引数の個数>][<標準値>]{<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。

`\newcommand`/`\newenvironment` 以外にも便利なコマンドがあります。

```
\providecommand{<命令>}[<整数>][<標準値>]{<定義>}
\DeclareRobustCommand{<命令>}[<数値>][<標準>]{<設定>}
```

この `\providecommand` はすでに命令が定義済みならば何もせず、もし定義されていないならば指定された内容の通りに命令を定義する事ができます。`\DeclareRobustCommand` を使った場合は頑丈な命令を作成できます。

▶ **問題 4.1** 次のようにアスタリスク付きで定義したコマンドと付与していない場合のコマンドの違いを考察してください。

```
\newcommand\testargs[1]{[#1]}
\newcommand*\testArgs[1]{[#1]}
\testArgs{1行目の入力\par 2行目の入力}.
\testargs{1行目の段落\par 2行目の段落}.
```

実際にタイプセットするとエラーが表示されます。これによりアスタリスク付きで定義した場合は複数行の段落を引数として受けとらなくなると考えられます。

▷ **例題 4.2** 中央揃えを行い文章を強調するような `cemph` 環境は次のように定義します。

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

▶ **問題 4.3** `\DeclareRobustCommand` を使った例を示します。次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。

```
\documentclass{jarticle}
\DeclareRobustCommand{\joubuyen}{\oalign{Y\cr cr\hss=\hss}}
\newcommand{\moroiyen}{\oalign{Y\cr cr\hss=\hss}}
\begin{document}
\tableofcontents
\section{もろい{\moroiyen}は壊れやすい}
どうですかねえ。
\section{丈夫な{\joubuyen}は壊れにくい}
大丈夫ですか？
\end{document}
```

この例では1回目のタイプセットでエラーが表示されます。

```
「
! Illegal parameter number in definition of \reserved@a.
<to be read again> \cr cr
1.6 \section{もろい{\moroiyen}は壊れやすい}
?
」
```

さらに2回目のタイプセットでは目次ファイル $\langle file \rangle$.tocが作成され、それが読み込まれるのでさらにエラーとなります。 $\langle file \rangle$.tocと $\langle file \rangle$.auxに何か記述されているのを吟味してください。 $\langle file \rangle$.auxには次のように出力されており、凄いなっています。

```
\relax
\@writefile{toc}{\contentsline{section}{\numberline{1}もろい
  {\lineskiplimit -\maxdimen \unhbox \voidb@x \vtop {\baselineskip
  \z@skip \lineskip .25ex\everycr{}}\tabskip \z@skip \halign {##\crcr
  Y\crcr \hss =\hss \crcr }}}は壊れやすい}{1}}
\@writefile{toc}{\contentsline{section}{\numberline{2}丈夫な
  {\joubyuen}は壊れにくい}{1}}
```

さらに $\langle file \rangle$.tocには次のような出力になります。

```
\contentsline{section}{\numberline{1}もろい{\lineskiplimit
-\maxdimen \unhbox\voidb@x \vtop {\baselineskip \z@skip
\lineskip .25ex\everycr{}}\tabskip \z@skip \halign {####\crcr
Y\crcr \hss =\hss \crcr}}}}は壊れやすい}{1}
\contentsline{section}{\numberline{2}丈夫な{\joubyuen}
は壊れにくい}{1}
```

これが`\DeclareRobustCommand`で`\moroiyen`命令を定義しなかった結果となります。L^AT_EXの命令を何か別のファイルに書き出す場合は丈夫な命令として定義していた方が無難なようです。ただし、次のように`\protect`命令を使った場合はうまく行く事を確認してください。

```
\section{もろい{\protect \moroiyen}は壊れやすい}
```

`\protect`を使うと`\moroiyen`が壊れずに $\langle file \rangle$.tocや $\langle file \rangle$.auxに出力されている事も確認してください。

▼ 4.2.4 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。例えば日本語では

「公園の中のベンチのうえ」

という名詞があったとすると、これはそれぞれ

「公園」「の」「中」「の」「ベンチ」「の」「うえ」

に分けて文を解釈し、**名詞**や**助動詞**などの**品詞**に分ける事ができます。

もう一つの例としてメールアドレスの場合を考えて見ましょう。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」を表す事を示す。

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事です。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 でも同じような事をやっています。

では文字列としてのコマンドが L^AT_EX でどのように解釈されるのかを考えてみましょう。ここでは引数を取る命令を考えてみます。まずは次のような入力の出力を予想してみましょう。

```
\newcommand{\OneArg}[1]{あ, #1だよ}
\OneArg ほあれそれ。
```

結果は「あ, ほだよあれそれ。」です。この結果から L^AT_EX では引数には何も指定がなければ 1 文字しか受け取らないという事が分かります。

続いて次のようにしてみましょう。

```
\newcommand{\OneArg}[1]{あ, #1だよ}
```

```
\OneArg {あれそれ}, それ.
```

結果は「あ、あれそれだよ、それ。」になるでしょう。どうやら引数を波括弧「{ }」で囲むと**それを一つの文字の塊として**受け取るようです。

次に二つ引数を取る命令を定義します。

```
\newcommand{\twoArgs}[2]{あ, #1だよ, ほら#2}
\twoArgs {それ} ね.
```

この場合の結果は「あ、それだよ、ほらね。」となるでしょう。もうお分かりですね。L^AT_EX の引数に複数の文字列を渡したいときは括弧でグルーピングすれば良いのです。さて今度は (T_EX の世界では通常の) 文字ではない '0' を含む命令を定義してみましょう。

```
\newcommand{\two@args}[2]{あ, #1だよ, ほら#2}
\two@args {それ} ね.
```

これをタイプセットするとエラーになるので Enter キーを 4 回ほど押すとエラーが表示されるでしょう。

```
「 ! Missing number, treated as zero.
<to be read again>
      g
1.7 \newcommand{\two@args}[2]{あ, #1だよ, ほら#2}
?
! You already have nine parameters.
\reserved@a ...def \expandafter \h \reserved@b #10
      g{
1.7 \newcommand{\two@args}[2]{あ, #1だよ, ほら#2}
?
! You can't use 'macro parameter character #' in horizontal mode.
<argument> あ, ##
      1だよ, ほら##2
1.7 \newcommand{\two@args}[2]{あ, #1だよ, ほら#2}
?
! You can't use 'macro parameter character #' in horizontal mode.
<argument> あ, ##1だよ, ほら##
      2
1.7 \newcommand{\two@args}[2]{あ, #1だよ, ほら#2}
?
」
```

これは一体どういう事でしょうか。メールアドレスの例を思い出してほしいのですが、L^AT_EX ではどうやら '0' をコマンドの区切りとして解釈しているようです。

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

```
\two@args {それ} ね.
```

そして引数の受け取り方を考えた場合の結果は「あ, @だよ, ほら ge それね。」となる事でしょう。

この事から L^AT_EX においての命令の定義には英字のみにする事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事です。

この文字の分類を利用して L^AT_EX 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次のように \two@args が定義できるわけです。

```
\newcommand{\two@args}[2]{あ, #1だよ, ほら#2}
```

そして \two@args の定義がマクロの中では可能なので、ユーザから \twoArgs 命令の実態を隠す事ができます。

```
\newcommand{\two@args}[2]{あ, #1だよ, ほら#2}
\newcommand{\twoArgs}{\two@args}
```

これは開発者と使用者の分離を行うために仕方がないのですが、逆に使用者が自分の思い通りに命令をカスタマイズできない等、様々な不満があるようです。

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事、 '@' 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.)
```

'@' を含むコマンドに関しては上記の二つの命令を使います。この命令の中身を見ても実は次のようになっています。

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

どうやら '@' のカテゴリーコードというものを 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カテゴリーコードと呼びます (表 4.1 参照)。

表 4.1 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_EX 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	十, 【 など

*1 標準では割り当てられていない

そのため何かマクロの中のコマンドに自分で変更を加えたいときは '@' を含む箇所を `\makeatletter` と `\makeatother` で囲んであげます。

```
\documentclass{jsarticle}
\makeatletter
\newcommand{\two@args}[2]{あ, #1だよ, ほら#2}
\newcommand{\twoArgs}{\two@args}
\makeatother
\begin{document}
\twoArgs{それ}{ね}.
\end{document}
```

▷ **例題 4.4** `\@ifstar` は '@' を含む命令で、引数の始めにアスタリスク (星) '*' があるかどうかを判定します。次の例の実行結果を吟味してください。

```
\newcommand\checkstar{\@ifstar{星付き}{星なし}}
\checkstar, \checkstar*, \checkstar!.
```

上記の例を実行すると次のようなエラーが表示されます。

```
[ ! You can't use '\spacefactor' in vertical mode. ]
```

やはり '@' を含む命令を用いるには `\makeatletter` と `\makeatother` で囲む必要があるようです。以下のように修正し、その実行結果を吟味してください。

```
\makeatletter
\newcommand\checkstar{\@ifstar{星付き}{星なし}}
\checkstar, \checkstar*, \checkstar!.
\makeatother
```

▼ 4.2.5 コマンドの引数

引数を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コマンド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}      a! b? とかはこだて! 未来? とか、さらに
{未来}とか、さらに\twoarg{\LaTeX}    \LaTeX! \LaTeX 2ε?
{\LaTeXe}
```

どうやら引数を取るコマンドに対してさらにコマンド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                  \LaTeX! \LaTeX 2ε? \LaTeX! 2? \LaTeX3
\twoarg\LaTeX2\LaTeX3
```

これは 4.2.4 節で行った内容が含まれています。'LaTeX' 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。

4.3 グループ・入れ子構造

TeX/LaTeX は変数のスコープ (有効範囲) というプログラミングでは当たり前の機能を持っています。これはプログラミングをやった事のない人には馴染みの薄いものなので詳しく説明しましょう。

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。

方言のように地方によってアクセントや言葉が違う事がありますが、L^AT_EX でも地方（範囲）を分けて言葉（定義）の使い分けができます。以下の例題をご自分で試してみてください。

```
\newcount\test% 新規に変数の調達
\the\test% 変数の値を表示
{% グループ 1の開始
  \test=3 \the\test% \test に3を代入して値を表示
  {% グループ 2の開始
    \test=5 \the\test% \test に5を代入して値を表示
  }% グループ 2の終了
  \the\test% グループ 2で test に5を代入しても?
}% グループの終了
\the\test% ここでの値は?
\test=6 \the\test% \test の値は?
```

`\newcount` は新規にカウンタ（レジスタ）を用意し、`\the` はカウンタなどの値を表示する命令です。さて、以上の入力から「035306」という値を得る事ができたでしょうか。この事を考えると「内側の括弧で代入した値は外側の括弧の領域に影響しない。」という事であり、「外側で値を変更しても内側の括弧にある値まで変更できない」という事です。まさに「グループ」を作ってその中で好き勝手にやっているわけです。

次に書体変更の宣言でどのように書体に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` という三つの宣言を使います。

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

ここでおやっ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まで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}          roman tt it tt it roman
tt \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` を使うと良い事になります。

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

▼ 4.3.1 大域化

書体の属性を変更する宣言型のコマンドはスイッチの変更をしている特殊なものでした。ここでは最初の大域変数と局所変数の二つをもう一度見直しておきましょう。

括弧の内側で定義した内容を外側でも有効にしたいときはどうすれば良いでしょう。また括弧の外側の定義を内側にも有効にする方法がないと不便でしょう。「用紙サイズ」のようにそう簡単に変わってほしくない変数は括弧の内側でも有効であってほしいものです。それらを括弧の内側でも有効にする事を**大域化**（グローバル化）と言います。TEX/L^AT_EX においても大域化のための `\global` という命令が用意されています。

```
\newcount\test \the\test
{%グループ 1
  \global\test=3 \the\test% 大域に代入
{%グループ 2
  \the\test
  \test=5 \the\test
}
\the\test
}
\the\test
\test=6 \the\test
```

とした場合は '0 3 35 3 36' となるでしょうか。3行目で大域的に '3' を代入していますので内側の括弧にも有効ですし、グループ 2 を出た後の数値も '3' です。そしてグループ 1 を抜けた後でも '3' が代入されています。以上のように `\global` を使うとそのグループの内側と外側の全ての処理に影響します (L^AT_EX において、いくつかの数値を操作する命令は、あらかじめ大域化するように定義されています)。

4.4 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。一つは始めと終わりをグループにする事で、どこからどこまでが中央揃えなのかが分かっているのでしょうか。

`\begin{center}`’ によってグループが始まり、`\end{center}`’ によってグループの終わりが示されています。

「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事に気づくでしょう。`\centering` 命令を使うと次のようになります。

```
{\centering A long long ago, a man was doing any test...}
```

しかし、非常に長い文章の場合は `\centering` 命令を使うよりは、`center` 環境としてグループ化したほうが分かりやすいでしょう。

```
\begin{center}
A long long ago, a man was doing any test...
\end{center}
```

そう考えるとコマンドには次の二つがある事が分かります。

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。

例として命令型の `\textsf` と宣言型の `\sffamily` を考えてみましょう。命令型の場合は次のような使い方はできません。

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

ただし、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. Roman?
This is sans serif.
\end{sffont}
\par 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.}
Roman!                                Roman! sans serif. 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。補足的な事です。

```
\begin{(環境名)} (要素) \end{(環境名)}
```

というのは L^AT_EX 側で ‘{<コマンド> <始めの処理> (要素) <終わりの処理>}’ に解釈されるので `\sffamily` のような宣言も

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

4.5 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` 命令で次のように使います。

```
<参照したい要素>\label{<ラベル名>}
```

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。

<code>\ref{ラベル}</code>	(通し番号)
<code>\pageref{ラベル}</code>	(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` の行頭のコメントは外してください)。

```
%\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
ちらを参照されたい。

詳しくは 104 ページの 4.5 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事も必要です。

▼ 4.5.1 相互参照の仕組み

節 (見出し) や図表には通し番号を付けます。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。ユーザが特にこの事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 4.2 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「参照用の文字列」の三つの要素を持っています。例えば `\newcounter{section}[chapter]` というのは、おおよそ次のような処理と同じ事になります (`\newcounter` とは新しいカウンタを定義するための命令です.)。

```
\newcount\c@section %素の番号用
\def\thesection{\thechapter.\c@section}%表示用
\def\p@section{\thechapter.\c@section}%参照用
```

この仕組みについて理解するには実際の動作を見るのが早いと思います。ファイル名 `reftest.tex` というファイルを作成し、1 回だけタイプセットしてください。

表 4.2 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

```
\documentclass{jsarticle}
\begin{document}
\newcounter{test} \thetest
\refstepcounter{test} \thetest
\end{document}
```

ここで `\refstepcounter` はカウンタの値を一つ増やす命令で `\thetest` はカウンタ `test` の値を表示するための命令だと思ってください。結果として端末には `No file refptest.aux` というメッセージが表示されるはずですが、この段階で `refptest.aux` を `cat` か `type` コマンドで見ると次の 1 行しか出力されていません。

```
\relax
```

`\refstepcounter` 命令だけでは参照できる状態にはないようです。

```
\refstepcounter{test} \thetest
```

上記の 1 行に対して次のように `'\label{cnt:test}'` を書き足して、1 回だけタイプセットを行ってください。

```
\refstepcounter{test} \thetest \label{cnt:test}
```


Label(s) may have changed. Return to get cross-ferencens right.

端末にはこのような L^AT_EX の警告が表示されます。ラベルが変更されたと思われるので解消しなさいと言われていました。ここで `reftest.aux` を見ると `\newlabel` 命令という新しい情報が出力されています。

```
\relax
\newlabel{cnt:test}{{1}{1}}
```

これで `cnt:test` という名前のラベルを参照する準備ができています。

```
\newcounter{test} \thetest
\refstepcounter{test} \thetest \label{cnt:test}
```

上記の 2 行に対して `\ref` と `\pageref` を含むような、次の 1 行を付け足して 1 回だけタイプセットします。

```
ref=\ref{cnt:test}, page=\pageref{cnt:test}
```

するとコンソールに警告は表示されません。`reftest.aux` の内容も変わっていません。さて、ここで L^AT_EX に意地悪をするため、`\setcounter` でページ番号 (`page`) を '100' にしてから 1 回だけタイプセットするとどうなるでしょうか。

```
\setcounter{page}{100}
\newcounter{test} \thetest
\refstepcounter{test} \thetest \label{cnt:test}
ref=\ref{cnt:test}, page=\pageref{cnt:test}
```

再び端末には

Label(s) may have changed. Return to get cross-ferencens right.

という警告が表示されてしまいました。そして `reftest.aux` のファイルの中身は「ページ番号は '100' である」という情報を含むように変更されています。

```
\relax
\newlabel{cnt:test}{{1}{100}}
```

以上の結果から分かるように `<file>.aux` には相互参照の情報が保存されている事が分かりました。L^AT_EX ではそれらを前回のタイプセットの結果が保存されていた `<file>.aux` と新しい相互参照の情報を比較してユーザに対しても警告を出しているという事が分かります。`\ref` 命令と `\pageref` 命令は相互参照用の情報からカウンタ番号やページ番号をラベルによって知ることができるという事です。

▼ 4.5.2 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 **L^AT_EX 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。

例えば jsbook クラスで章 (\chpater) の下の階層の節 (\section) 用のカウンタを新設するには \newcounter 命令を使って次のようにします。

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

このようなカウンタの操作には次の命令が使えます。

<code>\newcounter{<カウンタ名>}[<親カウンタ名>]</code>	(カウンタを新設する)
<code>\setcounter{<カウンタ名>}{<数値>}</code>	(<数値>を設定する)
<code>\addtocounter{<カウンタ名>}{<数値>}</code>	(<数値>を足す)
<code>\stepcounter{<カウンタ名>}</code>	(一つ足す)
<code>\refstepcounter{<カウンタ名>}</code>	(相互参照も可能に一つ足す)
<code>\value{<カウンタ名>}</code>	(値を表示する)

\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を変更するものに以下があります。

<code>\arabic</code>	(1, 2, ...)	<code>\roman</code>	(i, ii, ...)
<code>\Roman</code>	(I, II, ...)	<code>\alph</code>	(a, b, ..., z)
<code>\Alph</code>	(A, B, ..., Z)	<code>\fnsymbol</code>	(*, †, ...)

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}}
```

▶ **問題 4.5** 以下のファイルをタイプセットし、その実行結果を吟味してください。

```

\documentclass{jsarticle}
\renewcommand\thesection{\Alph[section]}
\begin{document}
\tableofcontents
\section{序論}
\subsection{構成}
\end{document}

```

この結果から、`\thesection` のみを変更するだけで十分かどうか検討してください。

4.6 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 4.3 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。

表 4.3 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)

簡単な例として節見出しを参照するときは次のような入力になります。

```

\section{加法混色}\label{sec:addmix}
それは、云々。
\section{減法混色}\label{sec:submix}
\ref{sec:addmix}~節\pp{\pageref{sec:addmix}ページ}では云々。

```

これは表 4.3 の規則にしたがって何のマクロも作成せず手動でやるとちょっと大変な事になります。

```

\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) では云々。

```

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) では云々。

表 4.3 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*\chapl原因{1}{\label{chap:#1}}
\newcommand*\chapref{1}{第~\ref{chap:#1}~章}
\newcommand*\seclab{1}{\label{sec:#1}}
\newcommand*\secref{1}{\ref{sec:#1}~節}
\newcommand*\figlab{1}{\label{fig:#1}}
\newcommand*\figref{1}{図~\ref{fig:#1}}
\newcommand*\tablab{1}{\label{tab:#1}}
\newcommand*\tabref{1}{表~\ref{tab:#1}}

```

```
\newcommand*\eqlab[1]{\label{equ:#1}}
\newcommand*\equiref[1]{式~\ref{equ:#1}}
```

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 $c_i$ は\equiref{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)では云々.
```

4

さて、最後の 1 行を見てみると \pageref において次のような記述が見受けられます。

```
\secref{addmixcolor}(\pageref{sec:addmixcolor}~ページ)では云々.
```

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*\pref[2]{(\pageref{#1:#2}ページ)}
\newcommand*\fullchapref[1]{第\ref{chap:#1}章 \pref{chap}{#1}}
\newcommand*\fullsecref[1]{\ref{sec:#1}節 \pref{sec}{#1}}
\newcommand*\fullfigref[1]{図~\ref{fig:#1} \pref{fig}{#1}}
\newcommand*\fulltabref[1]{表~\ref{tab:#1} \pref{tab}{#1}}
\newcommand*\fullequiref[1]{式~\ref{equ:#1} \pref{equ}{#1}}
```

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と思われます。

```
\section{減法混色}\seclab{submixcolor}
\fullsecref{addmixcolor}では云々.
```

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（本書の例を自分で入力するなど）、これを myref.sty としておくと便利でしょう*3。

*3 <http://tex.dante.jp/joul/myref.sty>

▼ 4.6.1 参照ラベルの表示——showkeys

`\label` と `\pageref` 及び `\ref` によって相互参照を行ないますが、参照するためのキーを原稿執筆段階で忘れてしまうことがあります。このようなときは `\label`、`\pageref`、`\ref` の参照されているラベルを出力してくれればありがたいものです。これには David Carlisle 氏による `showkeys` パッケージが使えます。次のようにすると、`\label` によって生成された `\newlabel` を傍注に出力し、`\ref`、`\pageref` で参照したラベルはその肩に付くようになります。

```
\usepackage{showkeys}
\section{序論}
\subsection{背景}\label{sec:back}
目標は\ref{sec:goal}~節を参照.
\subsection{目標}\label{sec:goal}
背景は\ref{sec:back}~節を参照. \par
改段落. \par
背景は~\pageref{sec:back}ページ.
```

sec:back

1.1 背景

目標は^{sec:goal}1.2 節を参照.

1.2 目標

背景は^{sec:back}1.1 節を参照.
改段落.
背景は^{sec:back}1 ページ.

sec:goal

▼ 4.6.2 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事を示します。

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事です。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ラベル名が定義されていない事になります。

Label(s) may have changed. Return to get cross-ferences right. が表示されたらラベルの値が変更されたという事なので、もう1度タイプセットをします。この作業は1度で終わらない事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事もあります。

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。

第 5 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。

5.1 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が明示的に空白を挿入する事もできます。
- 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事ができます。
- 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。

5.2 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。

▼ 5.2.1 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たときには `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 乗を足したものが

`c` の 2 乗に等しいという事は

`\(a^2 + b^2 = c^2 \)` と表せるが

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

`\end{math}` と書く事もできる。

a の 2 乗と b の 2 乗を足したものが c の 2 乗

に等しいという事は $a^2 + b^2 = c^2$ と表せるが

$a^2 + b^2 = c^2$ と書く事もできる。

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。

▼ 5.2.2 グルーピング

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ピングします。ここではべき乗を例にとって見てみましょう。

`\(a^{x+y} \neq a^{x+y} \)`

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

グルーピ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。

▼ 5.2.3 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。


```


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


```

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 ‘ $\left[\langle \text{数式} \rangle \right]$ ’ 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 $\left[c^2 = a^2 + b^2 \right]$ 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 $\begin{displaymath} a^2 + b^2 = c^2 \end{displaymath}$ と書く事もできます。

別行立て数式は $a^2 + b^2 = c^2$ と書く事もできます。

▼ 5.2.4 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* 環境で出力する事ができます。

```

\begin{equation}
\langle \text{数式} \rangle \label{\langle \text{ラベル} \rangle}
\end{equation}

```

equation で囲む事により 1 行の番号付きの数式を出力する事ができます。番号付きの数式は基本的にラベルを貼る事ができます。ラベルの参照の仕方は 4.5 節を参照してください。

$\begin{equation} a^2 + b^2 = c^2 \label{eq:equ} \end{equation}$ より $\$a^2+b^2\$$ は 式 (5.1) より c^2 は $a^2 + b^2$ に等しい。

$a^2 + b^2 = c^2$ (5.1)

▼ 5.2.5 複数行数式

```
\begin{eqnarray*}
<左辺> & \langle 関係子 \rangle & \langle 右辺 \rangle \\
<左辺> & \langle 関係子 \rangle & \langle 右辺 \rangle \\
\end{eqnarray*}
```

流れのある複数行の数式や証明などで関係子（例えば等号‘=’）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の行列に似ています。1行にはアンパサンド‘&’を二つまで、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事が可能です。

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

$$f(x) = x^2$$

$$f'(x) = 2x$$

▶ **問題 5.1** 次の入出力例を見て、それぞれの用法を確認してください。

```
\begin{eqnarray*}
5c = 4g \\
6a + 3c + 7d = 2g \\
3a + 5c = 3g \\
\end{eqnarray*}
```

$$5c = 4g$$

$$6a + 3c + 7d = 2g$$

$$3a + 5c = 3g$$

```
\begin{eqnarray*}
5c & = & 4g \\
6a + 3c + 7d & = & 2g \\
3a + 5c & = & 3g \\
\end{eqnarray*}
```

$$5c = 4g$$

$$6a + 3c + 7d = 2g$$

$$3a + 5c = 3g$$

```
\begin{eqnarray*}
& & 5c = 4g \\
& & 6a + 3c + 7d = 2g \\
& & 3a + 5c = 3g \\
\end{eqnarray*}
```

$$5c = 4g$$

$$6a + 3c + 7d = 2g$$

$$3a + 5c = 3g$$

```
\begin{eqnarray*}
& 5c = 4g & \\\
& 6a + 3c + 7d = 2g & \\\
& 3a + 5c = 3g & \\
\end{eqnarray*}
```

$$5c = 4g$$

$$6a + 3c + 7d = 2g$$

$$3a + 5c = 3g$$

▶ **問題 5.2** 次の入力例を処理すると、どのようなエラーが表示されるか確認してください。この結果からどのような事が言えるでしょうか。

```
\begin{eqnarray*}
& 5c & = & 4g & \\\
& 6a + 3c + 7d & = & 2g & \\\
& 3a + 5c & = & 3g & \\
\end{eqnarray*}
```

5

▼ 5.2.6 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 ‘\’ の前に貼ることができます。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事もできます。

```
\begin{eqnarray}
f(x) & = & x^2 & \label{eq1} \\
f'(x) & = & 2x & \label{eq2} \\
\int f(x)dx & = & x^3/3 + C & \nonumber
\end{eqnarray}
```

$$f(x) = x^2 \quad (5.2)$$

$$f'(x) = 2x \quad (5.3)$$

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

式~(\ref{eq1}) を微分したものが
式~(\ref{eq2}) である。

式 (5.2) を微分したものが式 (5.3) 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処理をしないでください。 `eqnarray*` 環境と同様に最終行に改行を入れしないでください。

5.3 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 5.1 の通りです。

表 5.1 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	<code>\mathnormal</code>	<i>ABCabc</i>
ローマン体	<code>\mathrm</code>	ABCabc
サンセリフ体	<code>\mathsf</code>	ABCabc
タイプライタ体	<code>\mathtt</code>	ABCabc
ボールド体	<code>\mathbf</code>	ABCabc
イタリック体	<code>\mathit</code>	<i>ABCabc</i>
カリグラフィック体	<code>\mathcal</code>	<i>ABC</i>

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

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

行列を表現するのにブラックボードボールド体（黒板風書体）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ルド体よりも行列である事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（表 5.2）。

表 5.2 `amssymb` による数式書体の拡張

書体	命令	出力
フラクショナル体	<code>\mathfrak</code>	\mathfrak{ABCabc}
ブラックボードボールド体	<code>\mathbb</code>	\mathbb{ABC}
数式内テキスト	<code>\text</code>	ABC 数式です

```
\usepackage{amssymb}
$$ x \in \mathbf{R} \neq x \in \mathbb{R}
x \in \mathbf{R} \neq x \in \mathbb{R}
$$ f(x) = 1/(1 + g(x)), (x = 3)
\text{とする}$$
```

▷ **例題 5.3** 以下の入力例の修正点を考えて下さい。

```
\begin{displaymath}
\bigotimes^n x \stackrel{\text{ncopies of } x}{=} x \otimes (x \otimes (\dots \otimes x) \dots)
\otimes (\dots \otimes x) \dots)
\sim \{n \text{ copies of } x\}
\end{displaymath}
```

実際には '*ncopies of x*' においては 'copies of' というのは文章になるべきですから、`\text` コマンドを使うのが良いでしょう (ただし x や n は数式になります)。さらに `\dots` による点の表現をしていますが、明示的にここでは `\cdots` 命令を使う方法が考えられます。よって修正後の入出力例は次のようになります。

```
\begin{displaymath}
\bigotimes^n x \stackrel{\text{n copies of } x}{=} x \otimes (x \otimes (\dots \otimes x) \dots)
\sim \{\text{\$n\$ copies of \$x\$}\}
\end{displaymath}
```

5.4 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ことがあります。ユーザ側で空白を調節するため表 5.3 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が空白を入ると意味的に正しくなります*1。

*1 本来ならば、より詳しく `\,`, `\;`, `\;`, `\!` がそれぞれ `\thinmuskip`, `\medmuskip`, `\thickmuskip`, 負の `\thinmuskip` という定義と同等であり、数式においては特別な単位 '`\mu`' が使われているという事を理解するのが良いのですが、本書ではその部分まで言及しません。

表 5.3 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>\sq</code>	<code>dx\sq 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 (1 em)	<code>\quad</code>	<code>dx\quad dy</code>	$dx dy$
全角の2倍の空白 (2 em)	<code>\qqquad</code>	<code>dx\qqquad dy</code>	$dx dy$
負の小さい空白	<code>\!</code>	<code>dx\! dy</code>	$dx dy$

積分 ' \int ' や全微分 ' dx ' のあいだにはユーザが空白を入れると見映えがします。

$$\backslash [\int \int f(x) dx dy \neq \int \int f(x) dx dy]$$

▷ **例題 5.4** 複数の式を1行に列挙する場合、次のようにすると適切な空白が挿入されません。

特性数 c, v, e を考えるとき, $[c = f_1 + f_2, v = 3f_2 + f_3, e = 2f_1 + 3f_3]$	特性数 c, v, e を考えるとき, $c = f_1 + f_2, v = 3f_2 + f_3, e = 2f_1 + 3f_3$
---	---

これを考慮すると、' c, v, e ' の部分には適切な単語間空白を挿入するのが望ましいでしょうし、`\quad` 程度の空きをそれぞれの式のあいだに挿入するのが適切だと思います。

特性数 c, v, e を考えるとき, $[c = f_1 + f_2, \quad v = 3f_2 + f_3, \quad e = 2f_1 + 3f_3]$	特性数 c, v, e を考えるとき, $c = f_1 + f_2, \quad v = 3f_2 + f_3, \quad e = 2f_1 + 3f_3$
---	---

▷ **例題 5.5** 式のすぐ後に条件等を記述するとき、単に丸括弧を後につづけるよりも、`\quad` 程度の空きを挿入するのが良いでしょう。

$$\backslash [g = \frac{1}{5}a + \frac{1}{6}b \quad (a > 3, b > 5)] \quad g = \frac{1}{5}a + \frac{1}{6}b \quad (a > 3, b > 5)$$

さらに、条件が複数あるときは $\backslash_$ によりある程度の空白を挿入するのも良いでしょう。

5.5 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事になります。

▼ 5.5.1 添え字

LaTeX での添え字の入力は簡単です。

$\langle \text{数式要素} \rangle^{\langle \text{上付き文字} \rangle}$
 $\langle \text{数式要素} \rangle_{\langle \text{下付き文字} \rangle}$

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 5.4 で例を示しますので参考してください。添え字を付けるときに上付きと下付きの順番は関係ありません。添え字は何もないものに対しても添える事が可能です。表 5.4 でもその方法がとら

表 5.4 添え字の使い方の例

意味	命令	出力	意味	命令	出力
右上	$x^{\{a+b\}}$	x^{a+b}	左上	$\{\}^{\{a+b\}}x$	$a+b x$
右下	$x_{\{a+b\}}$	x_{a+b}	左下	$\{\}_{\{a+b\}}x$	$a+b x$
右上と右下	$x^{\{a+b\}}_{\{c+d\}}$	x_{c+d}^{a+b}	左上と左下	$\{\}^{\{a\}}_{\{b\}}x$	$\frac{a}{b}x$
右上の右上	$x^{\{a^{\{b\}}\}}$	x^{a^b}	左下と右下	$\{\}_{\{a\}}x_{\{b\}}$	$a x_b$

れています。

$\backslash(\{\}^{\{a+b\}}_{\{x+y\}}A^{\{a+b\}}_{\{x+y\}}\backslash)$ $\frac{a+b}{x+y}A_{x+y}^{a+b}$

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 \backslashsp と下付きの \backslashsb 命令を使うと事もできます。

$\backslash(A^4_3 \backslashneq A \backslashsp4 \backslashsb3 \backslash)$ $A_3^4 \neq A_3^4$

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} \\
&& \mathfrak{t} A \neq \mathfrak{t} A
\end{eqnarray*}
```

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

▼ 5.5.2 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*’ 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 5.5 の通りです。

表 5.5 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>lim inf</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>lim sup</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)} (二項演算子として)
\pmod{(文字列)}
```

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


▼ 5.5.3 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 5.6 の通りです。

表 5.6 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}{2x^3 + 5x^2 + 8x + 5}
\end{displaymath}
```

$$\frac{1}{g(x)} + \frac{1}{2x^3 + 5x^2 + 8x + 5}$$

`\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。

<code>\limits</code> (上下に付く) <code>\nolimits</code> (肩に付く)
--

`\limits` を添え字を行うコマンドの前に置くと添え字をされる記号の上下に添え字を表示します。 `\nolimits` はその反対の事をします。

<pre>\begin{eqnarray*} \sum\nolimits_{n_{k=0}}k & \& \neq & & \sum_{k=0}^n k \\ \int^b_a dx & \& \neq & & \int_a^b dx \\ \end{eqnarray*}</pre>	$\sum_{k=0}^n k \neq \sum_{k=0}^n k$ $\int_a^b dx \neq \int_a^b dx$
--	---

<pre>\begin{eqnarray*} \lim\nolimits_{n\rightarrow 0}n & \& \neq & & \lim_{n\rightarrow 0} n \\ \prod_{n_{i=1}}n & \& \neq & & \prod_{i=1}^n n \\ \end{eqnarray*}</pre>	$\lim_{n \rightarrow 0} n \neq \lim_{n \rightarrow 0} n$ $\prod_{i=1}^n n \neq \prod_{i=1}^n n$
---	---

▼ 5.5.4 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勝手に大きさが変わります。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事もできます。

<pre>\begin{displaymath} \left[\Big(x+y\Big) \right] \\ \end{displaymath}</pre>	$\left[(x+y) \right]$
--	------------------------

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 5.7 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 `\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 5.7 から記号を選ぶ事によって、左右の区切りの対を自

表 5.7 主な区切り記号

()	\rfloor	\lceil	\updownarrow	\lbrace
)	(\lfloor	\rceil	\Uparrow	\rceil
[]	\arrowvert	\lceil	\Downarrow	\lceil
]	[\Arrowvert	\lceil	\Updownarrow	\lmoustache *
{	}	\Vert	\backslash	\backslash	\rmoustache *
}	{	\vert	\rangle	\rangle	\lgroup *
		\uparrow	\langle	\langle	\rgroup *
\	\	\downarrow	\rangle	\rangle	\bracevert *

* 大型の区切り記号です。

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が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}
```

$$\left(\frac{1}{1 + \frac{1}{1+x}} \right)$$

```
\begin{displaymath}
\left\uparrow \int f(x) dx \right\downarrow + \left\left[ \int g(x) dx \right]
\end{displaymath}
```

$$\left\uparrow \int f(x) dx \right\downarrow + \left[\int g(x) dx \right]$$

自分で括弧の大きさを指定する事もできます。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 5.8)。

表 5.8 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\

```
\begin{displaymath}
\Bigl| \Bigl( \int f(x) dx \Biggr) \Bigg| \quad \left\| \left( \int f(x) dx \right) / \left( \int g(x) dx \right) \right\|
\Bigg/ \Bigg( \int g(x) dx \Bigg) \Bigg|
\Bigg|
\end{displaymath}
```

表 5.8 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 `\bigm` 類を、右側を区切る記号には `\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。

```
[ \left|
\left( \int f(x) dx \right)
\Bigg/ \left( \int g(x) dx \right)
\right| \ ] \quad \left\| \left( \int f(x) dx \right) / \left( \int g(x) dx \right) \right\|
```

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( \left\uparrow
\int f(x) dx + \int g(x) dx
\right. \right) \ ] \quad \left( \uparrow \int f(x) dx + \int g(x) dx \right)
```

▼ 5.5.5 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 `\[\]` 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}
成分_{11} & \dots & 成分_{1n} \\
\vdots & & \vdots \\
成分_{m1} & \dots & 成分_{mn}
\end{array}
```

というように m 行 n 列の行列を書きます。ここでアンパサンド ‘&’ 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}
```

この部分では 4 列あるならば次のようにします。

```
\begin{array}{|c|cr}
```

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 環境の中で指定できる列指定子は表 5.9 となります。array 環境は入れ子にする事も、行列の中に行列を書いたりする事もできます。

表 5.9 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
@{ <i>表現</i> }	表現を 1 列追加する
p{ <i>長さ</i> }	ある列の幅を直接指定する
*{ <i>回数</i> }{ <i>列指定子</i> }	回数分だけ <i>列指定子</i> を繰り返す

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**。

```
\[ \left( \begin{array}{*{2}{c}}
  a & b \\ c & d
\end{array} \right)
\left( \begin{array}{c}
  m \\ n
\end{array} \right) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} m \\ n \end{pmatrix} = \begin{pmatrix} am + bn \\ cm + dn \end{pmatrix}
\left( \begin{array}{c}
  am+bn \\ cm+dn
\end{array} \right) \]
```

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。

```
\[ f(x)= \left\{
\begin{array}{cl}
x & (x > 0)\
0 & (x = 0)\
-x & (x < 0) \end{array}
\right. \]
```

$$f(x) = \begin{cases} x & (x > 0) \\ 0 & (x = 0) \\ -x & (x < 0) \end{cases}$$

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 5.10)。罫線などの使い方は以下の例を見て理解してください。

表 5.10 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>	行をつなげて列指定子通りに要素を出力します

```
\begin{displaymath}
\begin{array}{llc} \hline
\multicolumn{3}{c}{f(x)} \\ \hline
g(x) & h(x) & i(x) \\ \cline{2-2}
j(x)+k(x)+l(x) & o(x) & p(x) \\ \end{array}
\end{displaymath}
```

$$\begin{array}{llc} \hline
\multicolumn{3}{c}{f(x)} \\ \hline
g(x) & h(x) & i(x) \\ \cline{2-2}
j(x)+k(x)+l(x) & o(x) & p(x) \\ \end{array}$$

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} (a \ b \ c)
```

$$\begin{pmatrix} x \\ y \end{pmatrix} (a \ b \ c)$$

`\bordermatrix` 環境の括弧では各成分を区切るにはアンパサンド ‘&’ を使い、行の終わりには ‘`\cr`’ 命令を使います。

```
\[ A=\ \bordermatrix{
  & 1 & 2 \cr
  1 & a & b \cr
  2 & c & d } \]
```

$$A = \begin{matrix} & 1 & 2 \\ 1 & a & b \\ 2 & c & d \end{matrix}$$

別の方法として David Carlisle 氏の `delarray` (delimiter array) パッケージを用いる事もあります。次のようにすると `\left(` (`\right)` を補った場合と同様の括弧付けになります。

```
\usepackage{delarray}
$\begin{array}{cc}
  a_{11} & a_{12} \\
  a_{21} & a_{22}
\end{array}$
```

$$\left(\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array} \right)$$

次のように場合分けのときにも使えます。

```
$f(x) =
\begin{array}{l}
  1 & \mathrm{if} \ x > 0. \\
  0 & \mathrm{if} \ x = 0. \\
 -1 & \mathrm{if} \ x < 0.
\end{array}$
```

$$f(x) = \begin{cases} 1 & \text{if } x > 0. \\ 0 & \text{if } x = 0. \\ -1 & \text{if } x < 0. \end{cases}$$

上記のようにしなくとも、新たに列指定子を宣言して、次のようにもできます。

```
\usepackage{delarray}
\newcolumntype{V}{>{${}1<{${}}
\begin{displaymath}
  f(x) =
\begin{array}{l}
  1 & \text{if } \$x > 0$. \\
  0 & \text{if } \$x = 0$. \\
 -1 & \text{if } \$x < 0$.
\end{array}
\end{displaymath}
```

$$f(x) = \begin{cases} 1 & \text{if } x > 0. \\ 0 & \text{if } x = 0. \\ -1 & \text{if } x < 0. \end{cases}$$

5.6 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が表示形式を変更するには表 5.11 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

表 5.11 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	$\frac{a}{b}$

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 とするほうがスマート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。

`\(f(x)\)` の不定積分 `\(\int f(x)dx\)`
`\(\displaystyle \int f(x)dx\)` は
`\LaTeX` では少し違うし、分数は
`\frac{a}{b}` と書くよりも `a/b`
と書くほうがスマートである。

$f(x)$ の不定積分 $\int f(x)dx$ と $\int f(x)dx$ は
`ETEX` では少し違うし、分数は $\frac{a}{b}$ と書くよりも
 a/b と書くほうがスマートである。

`\[`
`\frac{1}{1+\frac{1}{1+\frac{1}{1+x}}}` ■
`\neq \frac{1}{\displaystyle 1+`
`\frac{1}{\displaystyle 1+`
`\frac{1}{1+x}}` `\]`

$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1+x}}} \neq \frac{1}{1 + \frac{1}{1 + \frac{1}{1+x}}}$$

`\(\int^b_a f(x)dx \neq`
`\{\displaystyle \int^b_a g(x)dx\}`
`\)`

$$\int_a^b f(x)dx \neq \int_a^b g(x)dx$$

▷ **例題 5.6** 次の数式の例をタイプセットし、出力結果を吟味してください。

```
\begin{displaymath}
f(x) = \frac{
\frac{a-b}{x+z} +
\frac{a+b}{x+y}
}{
\frac{1}{x+a} + z
}
```

$$f(x) = \frac{\frac{a-b}{x+z} + \frac{a+b}{x+y}}{\frac{1}{x+a} + z}$$

では、次のように入力するためにはどのような入力になるか、考えてみてください。

$$f(x) = \frac{(a-b)/(x+z) + (a+b)/(x+y)}{(1)/(x+a) + z}$$

簡単に考えると、次のように記述できると思われます。

```
\begin{displaymath}
f(x) = \frac{(a-b)/(x+z)+(a+b)/(x+y)}{(1)/(x+a)+z}
\end{displaymath}
```

しかし、どうせならばもう少しマクロで対処したい所です。例えば `\sfrac` 命令を次のように新設し、自動的に丸括弧とスラッシュを補うようにします。

```
\newcommand\sfrac[2]{(#1)/(#2)}
\begin{displaymath}
f(x) = \frac{\sfrac{a-b}{x+z}+\sfrac{a+b}{x+y}}{\sfrac{1}{x+a}+z}
\end{displaymath}
```

すると、先ほどと同じように入力できます。

$$f(x) = \frac{(a-b)/(x+z) + (a+b)/(x+y)}{(1)/(x+a) + z}$$

しかし、どうやら一番最後の `\sfrac{1}{x+a}` に関してはうまくいっていないようです。これに関してはおまじないとして次のように記述するとごまかせるでしょう。

```

\makeatletter
\newcommand\sfrac{\@ifnextchar\bgroup
  \@sfrac \@@sfrac}
\newcommand\@sfrac[2]{(#1)/(#2)}
\newcommand\@@sfrac[2]{#1/(#2)}
\makeatother
\begin{displaymath}
f(x) = \frac{\sfrac{a-b}{x+z}+
  \sfrac{a+b}{x+y}}{\sfrac{1}{x+a}+z}
\end{displaymath}

```

$$f(x) = \frac{(a-b)/(x+z) + (a+b)/(x+y)}{1/(x+a) + z}$$

ここでは丸括弧を補わない分子に関して**波括弧でグルーピングしない**という事で判別しています。

5.7 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。

▼ 5.7.1 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 5.12, 小文字の変体文字は表 5.13, 大文字は表 5.14 となります。ギリシャ

表 5.12 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>	\omicron <code>\omicron</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>

小文字においてオミクロン ‘ \omicron ’ だけはアルファベットの ‘ \omicron ’ と同じため特別に記号が用意されていません。逆に ‘ $\backslash\omicron$ ’ 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 \omicron 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 5.13 ギリシャ小文字の変体文字

```
\varepsilon \vartheta \varpi
\rho \varsigma \varphi
```

表 5.14 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 ‘\0’ 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 \0 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} \\
\mathit{foo} & \neq & \mathrm{foo}
\end{eqnarray*}
```

$$A \neq A$$

$$F(x) + C \neq F(x) + C$$

$$foo \neq foo$$

▼ 5.7.2 関係子や演算子などの数学記号

表 5.15 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 5.16 二項演算子

\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>	\triangle <code>\triangle</code>	\odot <code>\odot</code>
$*$ <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 5.17 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 5.18 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 5.19 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{m+M}</code>	$\overbrace{m+M}$ <code>\overbrace{m+M}</code>
$\underline{m+M}$ <code>\underline{m+M}</code>	$\underbrace{m+M}$ <code>\underbrace{m+M}</code>
$\overleftarrow{m+M}$ <code>\overleftarrow{m+M}</code>	$\widehat{m+M}$ <code>\widehat{m+M}</code>
$\overrightarrow{m+M}$ <code>\overrightarrow{m+M}</code>	$\widetilde{m+M}$ <code>\widetilde{m+M}</code>

```
\begin{displaymath}
\overbrace{a+b+c+d+e+f+g}^{h+i+j+k} + \underbrace{l+m+n}_{o+p+q}
\end{displaymath}
```

表 5.20 矢印

\leftarrow <code>\leftarrow</code>	\longrightarrow <code>\longrightarrow</code>	\leftrightarrow <code>\leftrightarrow</code>
\Leftarrow <code>\Leftarrow</code>	\Longrightarrow <code>\Longrightarrow</code>	\Leftrightarrow <code>\Leftrightarrow</code>
\hookleftarrow <code>\hookleftarrow</code>	\hookrightarrow <code>\hookrightarrow</code>	\rightharpoonleft <code>\rightharpoonleft</code>
\leftharpoonup <code>\leftharpoonup</code>	\hookrightarrow <code>\hookrightarrow</code>	\Longleftarrow <code>\Longleftarrow</code>
\leftharpoonup <code>\leftharpoonup</code>	\rightarrow <code>\rightarrow</code>	\Uparrow <code>\Uparrow</code>
\longleftarrow <code>\longleftarrow</code>	\rightarrow <code>\rightarrow</code>	\Downarrow <code>\Downarrow</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)
```

表 5.21 特殊な数学記号

\aleph	∂	\perp	\natural
\hbar	∞	\angle	\sharp
\imath	\prime	\triangle	\clubsuit
\jmath	\emptyset	\forall	\diamondsuit
ℓ	∇	\exists	\heartsuit
\wp	$\sqrt{\quad}$	\neg	\spadesuit
\Re	$ $	\backslash	
\Im	\top	\flat	

$\forall x \forall y (P(x, y) \vee (f(x) \wedge g(x)))$

$e^{j\theta} = \Re\{e^{j\theta}\} + \Im\{e^{j\theta}\} = \cos\theta + j\sin\theta$

表 5.22 点

\dots \cdots \vdots \ddots

$(a_0 + a_1 + \dots + a_n) \neq \{a_0, a_1, \dots, a_n\}$

▷ **例題 5.7** \ldots や \cdots 以外に \dots という命令もあります。これは自動的に \ldots と \cdots を切り替えてくれる命令です。

$\{f_n\} = f_1, f_2, \dots, f_n$

時おり適切に選定されない場合がありますので、その場合は手動で対処します。

▼ 5.7.3 標準ではない数学記号——`latexsym`

TEX_ε 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 5.23 標準ではない数学記号

\mho	\Join	\Box	\Diamond
\leadsto	\sqsubset	\sqsupset	\lhd
\unlhd	\rhd	\unrhd	

5.8 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 4.2 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。具体的な例として `Prob` 環境と `Exe` 環境を次のようにプリアンプルに記述します。

```
\newtheorem{Prob}{問題}[chapter]
\newtheorem{Exe}{例題}
```

そうしておけば以下のように使えます。

```
\begin{Exe}\label{Hoge:ware}
この文書は難しいか。答えは簡単だ。
\end{Exe}
\begin{Prob}\label{Geho:yueni}
この文書は適切かどうか考えよ。
\end{Prob}
例題~\ref{Hoge:ware}より
問題~\ref{Geho:yueni}が導かれる。
```

▷ **例題 5.8** この文書は難しいか。答えは簡単だ。

▶ **問題 5.9** この文書は適切かどうか考えよ。

例題 5.8 より問題 5.9 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。

▼ 5.8.1 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 氏による `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` 命令と同じように環境を新設します。

```
\newtheorem{<環境名>}{<名前>}
```

章などの親カウンタに連動させたい場合は次のように〈親カウンタ〉を指定します。

```
\newtheorem{<環境名>}{<名前>}[<親カウンタ名>]
```

同系の環境を作成するときは既存の環境名も指定して定義します。

```
\newtheorem{<環境名>}[<同系の環境名>]{<名前>}
```

`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用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「例題 2.1, 参考 2.2, 問題 2.3」のような環境を作成したければ次のようにすると良いでしょう。


```

{\theorembodyfont{\normalfont}
\theoremheaderfont{\normalfont\bfseries}
\newtheorem{Exam}{例題}
\newtheorem{Refer}[Exam]{参考}
\newtheorem{Prob}[Exam]{問題}}

```

5.9 その他有益な事柄

まずはマクロと数式を組み合わせた簡単な例を紹介します。

```

\newcommand*\niji[3][\% [a]{b}{c}
\ensuremath{#1x^2+#2x+#3=0}}

```

```

\newcommand*\Niji[3][\% [a]{b}{c}
\ensuremath{x=\frac{-#2\pm%
\sqrt{#2^2-4#1#3}}{2#1}}}}

```

二次方程式 $\niji[a]{b}{c}$ の一般解は

```
\begin{displaymath}
```

```
\Niji[a]{b}{c}
```

```
\end{displaymath}
```

となる。 $\niji{6}{5}$ の場合は

```
\niji{6}{5}より、$x=1,5$となる。
```

二次方程式 $ax^2 + bx + c = 0$ の一般解は

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

となる。 $x^2 + 6x + 5 = 0$ の場合は $x^2 + 6x + 5 = 0$ より、 $x = 1, 5$ となる。

不定積分を表現したり定積分を表現したりする次の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 $\int x$ や $\iint xy$ など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}
```

```

\[ \intx{f(x)} + \inty{g(y)} +
\iintxy{h(x,y)} \]

```

$$\int f(x)dx + \int g(y)dy + \iint h(x,y)dx dy$$

あまり複雑な数式になるとマクロを書くよりも直接書いたほうが良いかも知れません。

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 $Q(x)$ を書く事ができます。

```
\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\}}
```

```
$P(x)=x^2+\pi$ として $Q(x)=e^x$
```

```
とすると{\my}の一般解 $y$ は \[
```

```
\mypq{(x^2+\pi)}{e^x}\] となる。
```

$P(x) = x^2 + \pi$ として $Q(x) = e^x$ とすると
 $dy/dx + P(x)y = Q(x)$ の一般解 y 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
方をすると良いでしょう。

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{\pdif}[3] [] {\frac%
```

```
\partial^{#1}{#2}{\partial^{#3}{#1}}}
```

```
\[ \pdif{f}{x}+\pdif[2]{f}{x} \]
```

$$\frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial x^2}$$

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{\pdif}[3] [] {\frac%
```

```
\partial^{#1}{#2}{\partial^{#3}{#1}}}
```

```
\[ \pdif[2]{f}{x} + \pdif{\sp2f}{xy} +
```

```
\pdif[2]{f}{y} \]
```

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial xy} + \frac{\partial^2 f}{\partial y^2}$$

∂ と $\frac{\partial}{\partial}$ をごちゃごちゃ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。

▼ 5.9.1 記号の積み重ね

イコール‘=’のうえに‘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。一つ目の引数を二つ目の引数のうえ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}}{=}
\langle x \defeq p(t)+q(t)+r(t) \rangle
```

$$x \stackrel{\text{def}}{=} p(t) + q(t) + r(t)$$

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}
```

▼ 5.9.2 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。`\ooalign` と `\crrc` 命令を組み合わせたうまくできます。

```
{\ooalign{⟨一つ目⟩\crrc⟨二つ目⟩}}
```

二つの記号の内では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次のような定義をしておくといいでしょう。

```
\newcommand{\cnot}[1]{\ooalign{/\crrc{\hss#1}\hss}}
```

スラッシュは全角を使っています。

```
\newcommand{\pile}[2]{%
  {\oalign{#1\crcr#2}}
\newcommand{\cpile}[2]{\oalign{#1
  \hss#1\hss}\crcr{\hss#2\hss}}}
```

```
\newcommand{\cnot}[1]{%
  \oalign{/\crcr{\hss#1\hss}}}
```

特性数 $\$ \pile Y = \$$ は定数 $\$ \cpile Y = \$$ の云々で
 あり、 $\cnot\{A\}$ は \pile/A とは別物なので
 ある。

特性数 $\$$ は定数 $\$$ の云々であり、 A は A と
 は別物なのである。

▼ 5.9.3 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事があるそうです。
 方法として

- `\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事があります。
`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事ができません。

```
\(\mathbf{\int^a_b f(x)dx} \neq\)
\boldmath \(\int^a_b f(x)dx \neq\)  $\int_b^a f(x)dx \neq \int_b^a f(x)dx \neq \int_b^a f(x)dx$ 
\unboldmath\(\int^a_b f(x)dx \)
```

`\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。

```
\(\mathbf{\int^a_b f(x)dx} \neq\)
\boldsymbol{\int^a_b f(x)dx}\neq  $\int_b^a f(x)dx \neq \int_b^a f(x)dx \neq \int_b^a f(x)dx$ 
\int^a_b f(x)dx \)
```

現在は `amsbsy` を使うよりも `bm` パッケージの `\bm` 命令を使うのが良いでしょう。

$$\backslash(\mathbf{\int^a_b f(x)dx} \ \backslash\neq \ \backslash\mathbf{\int^a_b f(x)dx} \ \backslash\neq \ \int_b^a f(x)dx \neq \int_b^a f(x)dx \neq \int_b^a f(x)dx$$

$$\backslash\int^a_b f(x)dx \ \backslash)$$

▼ 5.9.4 高さや幅を揃える

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。

$$\backslash[\ \overline{\sqrt{a} + \sqrt{b}} \ \backslash\neq \ \sqrt{\mathstrut{a}} + \sqrt{\mathstrut{b}} \ \backslash]$$

$$\sqrt{a} + \sqrt{b} \neq \sqrt{a} + \sqrt{b}$$

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。

$$\backslash[\ \sqrt{\int f(x)dx} + \sqrt{g} \ \backslash\neq \ \sqrt{\int f(x)dx} + \sqrt{g} \ \backslash \ g \ \backslash]$$

$$\sqrt{\int f(x)dx} + \sqrt{g} \neq \sqrt{\int f(x)dx} + \sqrt{g}$$

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で高さと深さを 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。

$$\backslash[\ \underbrace{a+b} + \underbrace{i+j} \ \backslash\neq \ \underbrace{\smash{a+b}} + \underbrace{i+j} \ \backslash]$$

$$\underbrace{a+b} + \underbrace{i+j} \neq \underbrace{a+b} + \underbrace{i+j}$$

▷ **例題 5.10** 高さ、幅、深さを擬似的に模倣する `\phantom` 命令によって、次のような整列が可能となります。

```

\newcommand\PN[1]{\phantom{\mbox{#1}}}
\begin{eqnarray*}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 & \PN{+a_{23}x_3} & \\
& \& b_1 & \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 & & a_{11}x_1 + a_{12}x_2 & = b_1 \\
& \& b_2 & \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 & \PN{+a_{22}x_2} + a_{33}x_3 & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 & = b_2 \\
& \& b_3 & \\
& & a_{31}x_1 & + a_{33}x_3 = b_3 \\
\end{eqnarray*}

```

ただし、プラス '+' の前に何らかの要素をおかないと、適切な空白が挿入されないため、`\mbox` 命令を補っています。

▼ 5.9.5 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。

```

\[ \frac{\frac{a}{b}}{c} \neq \frac{a/b}{c}
\]

```

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せん。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}
\frac{\frac{a-b}{c}}{d} \neq \frac{a-b}{c} \neq \frac{(a-b)/c}{d}
\end{displaymath}

```

```

\begin{displaymath}
(a+f(x))/(a-g(x)) \neq \bigl(a+f(x)\bigr)\bigl(a-g(x)\bigr)
\end{displaymath}

```

▷ **例題 5.11** 次の入力例を `\frac` を使わないスタイルに書き直してください。

ある定理により $f = \frac{1}{ef + ev}$ であるから、 $g = \frac{f + e - v}{2}$ と定義する。

ある定理により $f = \frac{1}{ef + ev}$ であるから、 $g = \frac{f + e - v}{2}$ と定義する。

括弧を補うだけですから、次のようになります。

ある定理により $f = 1/(ef + ev)$ であるから、 $g = (f + e - v)/2$ と定義する。

ある定理により $f = 1/(ef + ev)$ であるから、 $g = (f + e - v)/2$ と定義する。

▼ 5.9.6 場合分けなど

一つの式から解が複数に**場合分け**される場合 `\cases` 命令が使えますが `amsmath` の `cases` 環境のほうがうまく行くでしょう。

```
\begin{cases}
<要素1> \\ \langle 要素2 \rangle \\ \dots \\ \langle 要素n \rangle
\end{cases}
```

```
\( f(x) = \begin{cases}
\,x & \text{\quad}(x>0) \\
\,0 & \text{\quad}(x=0) \\
\,-x & \text{\quad}(x<0)
\end{cases} \)
```

$$f(x) = \begin{cases} x & (x > 0) \\ 0 & (x = 0) \\ -x & (x < 0) \end{cases}$$

他にも `\choose` のように要素を縦に並べて括弧を付ける命令があります。

```
\choose (丸括弧付き) \brack (角括弧付き)
\brace (波括弧付き) \atop (括弧なし)
```

`\choose` などは全体を波括弧で括ってあげると正しく出力できます。

```
\[ {a + b \brack x + y} \neq
{a + b \brace x + y} \neq
{a + b \atop x + y} \]
```

$$\left[\begin{matrix} a + b \\ x + y \end{matrix} \right] \neq \left\{ \begin{matrix} a + b \\ x + y \end{matrix} \right\} \neq \begin{matrix} a + b \\ x + y \end{matrix}$$

```
\begin{displaymath}
\frac{a+b}{x+y} \neq \binom{a+b}{x+y}
\end{displaymath}
```

$$\frac{a+b}{x+y} \neq \binom{a+b}{x+y}$$

▼ 5.9.7 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{fool}`は`\(fool\)`にはなりません $fool$ は $fool$ にはなりませんから
`[fool \neq \mathit{fool}]. \]` $fool \neq fool.$

‘fool’ という文字が全て数式中では変数と解釈され、それぞれ L^AT_EX が適切だと思い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。

`$5,000\times 10=50,000$円です. \par` $5,000 \times 10 = 50,000$ 円です。
`$5{,}000\times 10=50{,}000$円です.` $5,000 \times 10 = 50,000$ 円です。

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{(p - 1)! (q - 1)!}{p! q! r!} \neq`
$$\frac{(p - 1)!(q - 1)!}{p!q!r!} \neq \frac{(p - 1)!(q - 1)!}{p!q!r!}$$

`\frac{(p - 1)!\,(q - 1)!}{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}
\newcommand*\temptxt{Is this text
mode?!}
\textit{\temptxt}\
\(\mathit{\temptxt})\
\(\temptxt)\
\(\mathit{Is\ this\ text\
mode?!})
```

Is this text mode?!
Isthistextmode?!
Isthistextmode?!
Is this text mode?!

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。

▼ 5.9.8 行列の省略点

例えば、連立方程式を列挙するときがあるとします。

```
\begin{eqnarray}
a_{11}x_1 + a_{12}x_2 + \dots +
a_{1k}x_k & = & b_1 \nonumber \\
a_{21}x_1 + a_{12}x_2 + \dots +
a_{1k}x_k & = & b_2 \nonumber \\
\vdots & = & \vdots \nonumber \\
a_{n1}x_1 + a_{n2}x_2 + \dots +
a_{nk}x_k & = & b_n \nonumber
\end{eqnarray}
```

$$\begin{array}{r}
 a_{11}x_1 + a_{12}x_2 + \cdots + a_{1k}x_k = b_1 \\
 a_{21}x_1 + a_{12}x_2 + \cdots + a_{1k}x_k = b_2 \\
 \vdots = \vdots \\
 a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nk}x_k = b_n
 \end{array}$$

しかし、これだと適切な空白が挿入されていないので、正統な入出力とは言えないでしょう。そこで次のように修正します。

```
\begin{displaymath}
\begin{array}{*2}{c@{\:+\:}\:}%
@{\cdots\:+\:}\:c@{\:=\;}\:c}
a_{11}x_1 & a_{12}x_2 & a_{1k}x_k & \\
& & & b_1 \\
a_{21}x_1 & a_{12}x_2 & a_{1k}x_k & \\
& & & b_2 \\
\vdots & \vdots & \vdots & \\
& \vdots & & \\
a_{n1}x_1 & a_{n2}x_2 & a_{nk}x_k & \\
& & & b_n
\end{array}
\end{displaymath}
```

$$\begin{array}{r}
 a_{11}x_1 + a_{12}x_2 + \cdots + a_{1k}x_k = b_1 \\
 a_{21}x_1 + a_{12}x_2 + \cdots + a_{1k}x_k = b_2 \\
 \vdots + \vdots + \cdots + \vdots = \vdots \\
 a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nk}x_k = b_n
 \end{array}$$

ただし、これでは3行目にまで影響が及んでいるため、意図した結果になりません。そこで `\multicolumn` 命令を用いて次のように修正します。

数字だけに限らずに、全角の英字、ローマ数字、アラビア数字は \LaTeX において**文中でも数式中でも使わないのが基本**です。次のような入力も絶対にやってはいけない例です。

$$ij\$/ e = 1 / 3 + 1 / q - 1 / 2$$

$$ij\$/ e = 1 / 3 + 1 / q - 1 / 2$$

とにかく、全角の英数字は使わないという方針で執筆すれば、 $\text{\TeX}/\text{\LaTeX}$ は正しい空白の調整ができるようになります。

▼ 5.10.2 文中の数式の入力

文中に出てくる数式、例えば「関数 $f(x)$ において $x = 0$ では」という表記は誤りで、正しくは次のようにします。

関数 $f(x)$ において $x = 0$ では (×) \\
関数 $f(x)$ において $x = 0$ では (○) 関数 $f(x)$ において $x = 0$ では (×)
関数 $f(x)$ において $x = 0$ では (○) 関数 $f(x)$ において $x = 0$ では (○)

どんなに細かい部分でも、ある部分が数式である以上、 \LaTeX に対してそこが「数式である」という事を明示的に教えます。これにより、適切に数式中での書体が選択されます。

▷ **例題 5.12** 次の入力には記法上の誤りがあります。どこに誤りがあるか見つけてください。

n 次元の特性数 v 、 1 次元の特性数 e 、
 2 次元の特性数 f を考えるとき \dots n 次元の特性数 v 、 1 次元の特性数 e 、 2 次元の特性数 f を考えるとき \dots

入力例では文中のアラビア数字が全角になっています。さらに本来ならば、イタリック体になるべき英字も文章における通常のローマン体になっていますので、次のように修正します。

n 次元の特性数 v 、 1 次元の特性数 e 、
 2 次元の特性数 f を考えるとき \dots n 次元の特性数 v 、 1 次元の特性数 e 、 2 次元の特性数 f を考えるとき \dots

何らかの要素を列挙するときに、数式モードでのコンマ、を使うと、文中でのコンマの空白が入りません。そのため、次のような入力は避けた方が無難です。

特性数 v 、 e 、 f を考えるとき \dots 特性数 v, e, f を考えるとき \dots

文書全体における句読点に全角のコンマ・ピリオドか、それとも半角のカンマ・ピリオドか、または全角の点丸を使うかという問題もあります*3。投稿先によっては指定される場合もあります。一貫性を重視するのであれば、以下のように半角のピリオド・コンマで修正するのが良いでしょう。これにより適切な行分割も行われるようになります。

特性数 v , e , f を考えるとき
`\ldots`.

特性数 v, e, f を考えるとき....

5.11 数式表現の拡張—— $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$

米国数学会 (American Mathematical Society) が提供していた数式記述用のマクロ $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ を Frank Mittelbach 氏と Rainer Schöpf 氏が $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ に移植しました。これが $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ と呼ばれるマクロ群です。 $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ には数式記述に関するマクロパッケージ `amsmath` や、 $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ では用意されていない記号を提供する $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{O}\mathcal{N}\mathcal{T}\mathcal{S}$ などが含まれています*4。 $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ を用いる事で、表現の幅が広がります*5。

$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ はおおむね次のパッケージ群で構成されるツールです。

amsmath $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ の核となる種々のマクロを含むパッケージです。自動的に `amstext`, `amsgen`, `amsbsy`, `amsopn` のパッケージを読み込みます。

amsbsy 数式を太字にする `\boldsymbol` を使うための命令が定義されています。

amstext 数式中で文章を出力する `\text` 命令が定義されています。

amscd ダイアグラムを描くための CD 環境が定義されています。

amsopn 新規に演算子を定義するための `\DeclareMathOperator` 命令が定義されています。

amscd 可換図を描くためのパッケージです。

amsxtra `amsmath` パッケージからは外れている補助的なコマンドが定義されているパッケージです。

amssymb 種々の拡張的な記号が定義されているファイルです。 `amssymb` パッケージ

*3 句読点問題に関しては宗教論争にもなり得る話題であり、奥が深いため、本書では詳しく言及しません。

*4 $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ のバージョンは現在 2 で古いバージョン 1.2 がインストールされている場合は、更新するのが良いでしょう。

*5 表現の幅が広がるという事は一般的な原則から徐々に少しずつ離れていくという側面も多少なりあると思います。そのため、全ての著者が $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ の表現方法に納得できるとは限らないでしょう。

ジを自動的に読み込みます。

eucal 標準の Computer Modern のカリグラフィック体ではなく、Euler スクリプト体と呼ばれる書体に変更するためのパッケージ。

▶ **問題 5.13** `amsmath` パッケージのパッケージオプションとしていくつか指定できるものがあります。

数式番号に関連する `centertags`, `tbtags`, 添字に関連する `sumlimits`, `nosumlimits`, `intlimits`, `nointlimits`, `namelimits`, `nonamelimits`, 数式の揃え等に関連する `leqno`, `ceqno`, `fleqn` 等があります。実際にオプションを指定して、その挙動を確認してください。

<pre>\begin{equation} \begin{split} f &= a + b + c + d \\ &+ e + f + g + h \\ &= i + j \end{split} \end{equation}</pre>	$ \begin{aligned} f &= a + b + c + d \\ &+ e + f + g + h \\ &= i + j \end{aligned} \tag{5.4} $
---	---

<pre>\begin{multline} f = a + b + c + d + e \\ + g + h + i + j \\ + k + l + m + n \\ + o + p + q \end{multline}</pre>	$ \begin{aligned} f &= a + b + c + d + e \\ &+ g + h + i + j \\ &+ k + l + m + n \\ &+ o + p + q \end{aligned} \tag{5.5} $
---	--

<pre>\begin{gather} a = b + c \\ c = d + f + g \end{gather}</pre>	$ a = b + c \tag{5.6} $
<pre>\end{gather}</pre>	$ c = d + f + g \tag{5.7} $

<pre>\begin{align} a &= b + c + d \\ f(x) &= g + h + i + j \end{align}</pre>	$ a = b + c + d \tag{5.8} $
<pre>\end{align}</pre>	$ f(x) = g + h + i + j \tag{5.9} $

`multline` 1行目は左揃え、2行目から最後の一つ前まで中央揃え、最期の行が右揃えになります。明示的に `\shoveleft`, `\shoveright` で揃えの調整ができます。

整列付き環境 基本的には行列を記述するための環境です。

`matrix` 何列でも記述可能な行列で、列指定子を必要とせず、各成分は中央揃えされます。

`cases` 場合分けに使用できる環境で、波括弧が左側に補われます。何行でも記述可能です。

`array` `\hdotsfor` 命令が使えるように拡張されています。

位置合わせ付きの環境 以下の環境は何列でも何行でも記述可能です。

`align` 1行に複数の数式を記述するための環境です。奇数個目のアンパサンドが位置合わせに使われます。数式間には自動的に良い加減の空気が挿入されます。

`flalign` 1行に複数の数式を記述するという意味では `align` と同じですが、数式間には目一杯の空気が挿入されます。

`alignat` 自分で空きと列を指定する環境です。

`split` 他の数式環境の補助用として1行の数式を複数行に分割できます。

- 基本的にどの環境も最終行に改行 `\\` は必要ありません。
- 番号付きの数式環境、`gather`, `multline`, `align`, `flalign`, `alignat` はアスタリスク `*` を付ける事により、番号付けをしなくなります。
- 番号付けは `\tag` と `\notag` によって変更できます。 `\tag*` を使うと括弧なしで出力します。

▶ **問題 5.14** 各数式環境において、等号の後にアンドを置くと、タイプセット後の結果はどうなるか吟味してください。

```
\begin{align}
```

```
  a =& b + c + d \\
```

$$a = b + c + d \quad (5.15)$$

```
f(x) =& g + h + i + j
```

$$f(x) = g + h + i + j \quad (5.16)$$

```
\end{align}
```

結果的には適切な空白が挿入されていません。ですから、間違ってアンドを二項関係子の後に置かないように注意してください。

▼ 5.11.2 gather 環境

gather は 1 列で全ての数式が中央揃えになります。

▶ **問題 5.15** 中央揃えになるという事は、 $\text{L}^{\text{T}}\text{E}^{\text{X}} 2_{\epsilon}$ 標準の eqnarray 環境と同じような事ができるでしょうか。次の記述と gather 環境が等価であるか検証してください。もし違いがあるのであれば、どの部分に差異があるのか整理してください。

```
\begin{eqnarray*}
& a = b & \backslash
& c = b + d & \backslash
& d = d + e &
\end{eqnarray*}
```

▼ 5.11.3 split 環境

split 環境は一行では収まりきらないような長い数式を複数行に分割するときに使います。displaymath 命令 や equation 環境の中で使います。

```
\begin{displaymath}
\begin{split}
f(x) &= x^9 + \frac{1}{9}x^8 + \\
&\frac{1}{8}x^7 + \cdots \\
&+ \cdots
\end{split}
\end{displaymath}
```

$$f(x) = x^9 + \frac{1}{9}x^8 + \frac{1}{8}x^7 + \cdots + \cdots$$



実際には `\cdots` 命令以外にも $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{E}\mathcal{T}\mathcal{E}\mathcal{X}$ には `\dotsc` (コンマ), `\dotsb` (二項演算子か, 関係子), `\dotsm` (乗算), `\dotsi` (積分記号), `\dotso` (上記以外) の五つの三点リーダーが定義されています。これによりそれぞれの記号用に良い加減に調整された空白が挿入されるようになります。

級数 `a_1, a_2, \dotsc`
直和 `$a_1 + a_2 + \dotsb$`
直積 `$a_1 a_2 \dotsm$`
積分 `$$\int_{a_1}\int_{a_2}\dotsi$`

級数 a_1, a_2, \dots
直和 $a_1 + a_2 + \cdots$
直積 $a_1 a_2 \cdots$
積分 $\int_{a_1} \int_{a_2} \cdots$

▼ 5.11.4 align, flalign, alignat 環境

align 環境は 1 行に複数の数式を記述するための環境です。奇数個目のアンドが位置合わせに使われます。数式間には自動的に良い加減の空きが挿入されます。

```
\begin{align}
  a &= b + c + d \\
f(x) &= g + h + i + j
\end{align}
```

$$a = b + c + d \quad (5.17)$$

$$f(x) = g + h + i + j \quad (5.18)$$

```
\begin{align}
  a_1 &= b + c & g_1 &= d + e \\
  a_2 &= f + h & g_2 &= i + j
\end{align}
```

$$a_1 = b + c \quad g_1 = d + e \quad (5.19)$$

$$a_2 = f + h \quad g_2 = i + j \quad (5.20)$$

flalign 環境は 1 行に複数の数式を記述するという意味では align と同じですが、数式間には目一杯の空きが挿入されます。

```
\begin{flalign}
  a_1 &= b + c & g_1 &= d + e \\
  a_2 &= f + h & g_2 &= i + j
\end{flalign}
```

$$a_1 = b + c \quad g_1 = d + e \quad (5.21)$$

$$a_2 = f + h \quad g_2 = i + j \quad (5.22)$$

数式同士の空きを手動で調整するには alignat 環境を使います。

```
\begin{alignat*}{2}
(a+b)^2 &= a^2+2ab+b^2 & \\
\quad &\text{\text{展開する}} \\
&=a(a+2b)+b^2 & \\
&\text{\text{\$a\$ で括る}} & \\
\end{alignat*}
```

$$(a + b)^2 = a^2 + 2ab + b^2 \quad \text{展開する}$$

$$= a(a + 2b) + b^2 \quad a \text{ で括る}$$

▼ 5.11.5 multline

最初の行が左揃え、中間の行は中央揃え、最後の行が右揃えになります。明示的に \shoveleft と \shoveright 命令で揃えを変更できます。

```

\begin{multline}
f = a + b + c + d + e \\
+ g + h + i + j \\
\shoveleft{+ k + l + m + n} \\
+ o + p + q
\end{multline}

```

$$\begin{aligned}
f &= a + b + c + d + e \\
&+ g + h + i + j \\
&+ k + l + m + n \\
&+ o + p + q
\end{aligned} \tag{5.23}$$

▶ **問題 5.16** 以下のファイルをタイプセットし、その出力結果を吟味してください。

```

\setlength\multlinegap{10pt}
\begin{multline}
f = a + b + c + d + e \\
+ g + h + i + j \\
+ k + l + m + n \\
+ o + p + q
\end{multline}
\setlength\multlinegap{30pt}
\begin{multline}
f = a + b + c + d + e \\
+ g + h + i + j \\
+ k + l + m + n \\
+ o + p + q
\end{multline}

```

これにより `\multlinegap` の役割は何だと考えられるでしょうか。

▼ 5.11.6 括弧付の行列

$\mathcal{A}_{\text{M}}\text{S}_{\text{E}}\text{T}_{\text{E}}\text{X}$ では括弧の入力を省略するために、`matrix` 環境以外にも次の五つの環境が定義されています。

<code>pmatrix</code>	丸括弧 (\langle 行列要素 \rangle)	<code>bmatrix</code>	角括弧 [\langle 行列要素 \rangle]
<code>Bmatrix</code>	波括弧 { \langle 行列要素 \rangle }	<code>vmatrix</code>	縦棒 \langle 行列要素 \rangle
<code>Vmatrix</code>	二重縦棒 \langle 行列要素 \rangle		

```

\begin{math}
\begin{pmatrix}
a_{11} & a_{11} \\
a_{21} & a_{22}
\end{pmatrix}
\end{math}

```

$$\begin{pmatrix} a_{11} & a_{11} \\ a_{21} & a_{22} \end{pmatrix}$$

文中数式で `matrix` 環境を使うと $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ のようになります。本来ならば $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ となった方がよいと思いますので、この場合は `\smallmatrix` 環境を使います。

行列 `$A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)` に関しては、云々であるから `\ldots`。

行列 $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ に関しては、云々であるから....

▷ **例題 5.17** 5.9.8 節において行列の省略点を種々の方法で出力する試みをしましたが、 $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{I}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ には `\hdotsfor` 命令が用意されており、次のような使い方ができます。

```
\begin{displaymath}
\begin{array}{*2}{c@{\:+\:}}%
@{\cdots\:+\:}c@{\;=\;}c}
a_{11}x_1 & a_{12}x_2 & a_{1k}x_k \\
& & & & b_1 \\
a_{21}x_1 & a_{22}x_2 & a_{2k}x_k \\
& & & & b_2 \\
\hdotsfor{4} \\
a_{n1}x_1 & a_{n2}x_2 & a_{nk}x_k \\
& & & & b_n \\
\end{array}
\end{displaymath}
```

$$\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1k}x_k = b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2k}x_k = b_2 \\
\cdots \cdots \cdots \cdots \cdots \cdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nk}x_k = b_n
\end{array}$$

`\hdotsfor` の点の間隔は '`\hdotsfor[<点の数>]{<列数>}`' のように任意引数で指定できます。

▶ **問題 5.18** 5.9.1 節の例では `\substack` 命令を使って積み重ねを行っていましたが、`amsmath` には他にも `subarray` 環境も用意されています。そこで、以下の入出力を吟味してください。

```

\begin{displaymath}
\sum_{\substack{i\le 1\le l \\ j\le 1\le m \\ k\le 1\le n}} p_i q_j r_k \neq \sum_{\substack{i\le 1\le l \\ j\le 1\le m \\ k\le 1\le n}} p_i q_j r_k
\end{displaymath}

```

これにより、`\substack` 命令と `subarray` 環境の決定的な違いは揃えの位置を決められるかどうかになります。`\substack` は有無を言わずに中央揃えになる事でしょう。

▼ 5.11.7 数式番号の工夫

通常、数式番号はそれぞれの式に対して一意に番号が振られます。

```

\begin{align}
(a+b)^2 &= a^2+2ab+b^2 \\
&= a(a+2b)+b^2
\end{align}

```

$$(a+b)^2 = a^2 + 2ab + b^2 \quad (5.24)$$

$$= a(a+2b) + b^2 \quad (5.25)$$

しかし、ほとんど同義の数式をグループにしたい時があり、次のように手動で `\tag` 命令を使ったとします。

```

\begin{align}
g &= (a+b)^2 \quad \label{eq:x} \\
&= a^2+2ab+b^2 \quad \tag{\ref{eq:x}a} \\
&= a(a+2b)+b^2 \quad \tag{\ref{eq:x}b}
\end{align}

```

$$g = (a+b)^2 \quad (5.26)$$

$$= a^2 + 2ab + b^2 \quad (5.26a)$$

$$= a(a+2b) + b^2 \quad (5.26b)$$

これでは親の数式がなければうまくいきません。そこで `subequations` 環境と呼ばれる専用の環境を使います。


```

\newcommand*\END{\mathop{\mathrm{End}}}
\newcommand*\MK{\mkern-4mu}
\newcommand*\Leq{\hbox to 3em{\$=\MK=\MK=\MK=\MK=$}}
\[ \begin{array}{ccc}
S^{\{\mathcal{W}\}}_{\Lambda} \otimes T & \& \law{\sim} & T \\
\lar \downarrow & \& \& \rar \uparrow
\end{array} \]
(S \otimes T) / I & \& \& (Z \otimes T) / J
\end{array} \]

```

▼ 5.11.9 追加された演算子等

表 5.24 amsmath で追加されたギリシャ大文字の変体文字

Γ	\varGamma	Λ	\varLambda	Σ	\varSigma	Ψ	\varPsi
Δ	\varDelta	Ξ	\varXi	Υ	\varUpsilon	Ω	\varOmega
Θ	\varTheta	Π	\varPi	Φ	\varPhi		

表 5.25 amsmath で追加された数学関数

injlim	$\operatorname{projlim}$	$\operatorname{varliminf}$
$\operatorname{varlimsup}$	$\operatorname{varinjlim}$	$\operatorname{varprojlim}$

表 5.25 にも定義されていない独自の数学関数を新規に定義するのであれば、プリアンブルで `\DeclareMathOperator` 命令が使えます。

```
\DeclareMathOperator*{関数名}{定義内容}
```

星を付けると `\limits` を伴った事と等価になります。

▶ **問題 5.19** 以下のファイルをタイプセットし、その実行結果を吟味してください。

```

\documentclass{jsarticle}
\usepackage{typelcm,amsmath}
\newcommand*\End{\mathop{\mathrm{End}}}
\DeclareMathOperator{\END}{\mathrm{End}}
\begin{document}
\begin{align}
\int \mathrm{End} x \, dx &= cx \\
\int \End x \, dx &= bx \\
\int \END x \, dx &= ax
\end{align}

```

```
\end{align}
\end{document}
```

何かしらの新規の数学関数を定義する場合、`\DeclareMathOperator` 命令を使うのが良い事になります。一部分にしか使わない場合は `\operatorname` 命令が使えます。

表 5.26 `amsmath` で追加された積分記号

```
∫ \oint ∬ \iint ∭ \iiint
∫∫∫ \iiiint ∫⋯∫ \idotsint
```

```
\begin{align*}
\int\int f(x,y)\,dx\,dy = g(x,y) \quad \iint f(x,y) \,dx \,dy = g(x,y)
\int\!\!\!\int f(x,y)\,dx\,dy = g(x,y) \quad \iiint f(x,y) \,dx \,dy = g(x,y)
\iint f(x,y)\,dx\,dy = g(x,y) \quad \int\int f(x,y) \,dx \,dy = g(x,y)
\end{align*}
```

表 5.27 で追加されたアクセントにおいて、`\ddddot` と `\dddotted` 以外は基本的に二重のアクセントを出力するために使われます。

表 5.27 `amsmath` で追加されたアクセント記号

```
¨ \ddddot{a}      ¨¨ \dddotted{a}      ˆ \Hat{\Hat{A}}
Á \Acute{\Acute{A}}    ¯ \Bar{\Bar{A}}      ˙ \Dot{\Dot{A}}
Č \Check{\Check{A}}    ˘ \Grave{\Grave{A}}    ˆˆ \Vec{\Vec{A}}
˙˙ \Ddot{\Ddot{A}}    ˘˘ \Breve{\Breve{A}}    ˜ \Tilde{\Tilde{A}}
```

表 5.28 `amsxtra` で追加された添字アクセント記号

上付き添字としてのアクセントですから ‘`A\spshat`’ のように使います。

```
Aˆ \spshat    A˘ \spcheck    A˜ \sptilde
A˙ \spdot    A¨ \spddot    A˙˙ \spdddot
A˘ \spbrev
```

表 5.29 amsmath で追加された空白命令

```

\thinspace ㊦ \negthinspace ㊦
\medspace ㊦ \negmedspace ㊦
\thickspace ㊦ \negthickspace ㊦

```

▼ 5.11.10 その他のコマンド

▶ **問題 5.20** `align`, `gather`, `alignat` は基本的に、その行いっばいに数式を出力する環境であるため、文中で使うというような事ができません。そこで `aligned`, `gathered`, `alignedat` 環境がそれぞれ用意されています。

```

\begin{equation*}
\left.
\begin{aligned}
I &= E/R \\
E &= RI
\end{aligned}
\right\} \text{ オームの法則}
\end{equation*}

```

`aligned` 環境等を文中で使用する時に任意引数に ‘t’, ‘c’, ‘b’ を指定するとどうなるか、実際に試してみてください。

▷ **例題 5.21** 何かしらの理由により、複数行の別行立て数式の途中に文章を挿入したい時があります。AMSTeX において途中に文章を入れるには `\intertext` 命令が使えます。

```

\begin{align*}
(a+b)^2 &= a^2 + 2ab + b^2 \\
\intertext{展開する} & \\
&= a(a+2b) + b^2 \\
\intertext{さらに \$a\$ で括る} & \\
\end{align*}

```




別行立て数式では `\` 命令によって数式を改行しますが、通常 `\` だけではページの区切りで分割されません。これを分割できるようにするには `\displaybreak` 命令を使います。`\displaybreak` は任意引数を取り、1-4 の数値を与える事により改ページのし易さを指定できます。プリアンブルに `\allowdisplaybreaks` を記述すると、文書全体において、別行立て数式における分割のし易さを指定できます。`\allowdisplaybreaks` も `\displaybreak` と同様に任意引数を取ります。`\allowdisplaybreaks` を指定しており、逆に分割されたくない時は `*` 命令を使います。

▷ **例題 5.22** \LaTeX には `\overrightarrow`, `\overleftarrow` という大きなアクセント記号がありますが、`amsmath` では、`\overleftarrow`, `\underleftarrow`, `\underrightarrow`, `\underletrightarrow` の四つが追加されています。

```
\begin{gather*}
\overrightarrow{a + b} \\
\underleftarrow{a + b} \\
\underrightarrow{a + b} \\
\underletrightarrow{a + b}
\end{gather*}
```

$$\begin{array}{l} \overrightarrow{a + b} \\ \underleftarrow{a + b} \\ \underrightarrow{a + b} \\ \underletrightarrow{a + b} \end{array}$$

さらに小さなサイズの添字を自動的に付ける `\xleftarrow` と `\xrightarrow` があります。

`\xleftarrow[下付き]{上付き}` `\xrightarrow[下付き]{上付き}`

```
[ A \xleftarrow{\alpha + 1} B
\xrightarrow[X]{\beta - 1} C \]
```

$$A \xleftarrow{\alpha + 1} B \xrightarrow[X]{\beta - 1} C$$

▷ **例題 5.23** 5.5.1 節において添字を出力する命令を紹介しました。さらに `\overset`, `\underset`, `\sideset` という三つの便利な命令が追加されています。`\overset` と `\underset` は添字サイズで上付き・下付き記号を付けられます。`\sideset` は機能的には `leftidx` パッケージの `\leftidx` と同じようなものです。

```
\begin{gather*}
\overset{*}{X} \neq \underset{*}{X} \\
\leftidx[_a^b]{\prod}_{c^d} \\
\sideset[_a^b]{c^d}{\prod}
\end{gather*}
```

$$\begin{array}{l} X^* \neq X_* \\ \prod_{c^d}^b \\ \prod_{c^d}^b \end{array}$$

▼ 5.11.11 分数の拡張

LaTeX 標準の `\frac` 以外にも、`\textstyle` を補う `\tfrac`、`\displaystyle` を補う `\dfrac` が用意されています。

$$\begin{aligned} \backslash \frac{R}{I} &\neq \backslash dfrac{R}{I} \\ &\neq \backslash tfrac{R}{I} \end{aligned} \qquad \frac{R}{I} \neq \frac{R}{I} \neq \frac{R}{I}$$

▷ **例題 5.24** `\frac` の場合と同様に `\binom` においても `\dbinom` と `\tbinom` 命令が用意されています。

$$\begin{aligned} \backslash \binom{k}{1} &\neq \backslash dbinom{k}{1} \\ &\neq \backslash tbinom{k}{1} \end{aligned} \qquad \binom{k}{1} \neq \binom{k}{1} \neq \binom{k}{1}$$

もっと一般的に、分母・分子の関係にあるような数式コマンドを定義するために `\genfrac` 命令があります。

`\genfrac{<左括弧>}{<右括弧>}{<線の太さ>}{<スタイル>}{<分子>}{<分母>}`

`<スタイル>` には 0-3 までの数字を指定し、それぞれ `\displaystyle`、`\textstyle`、`\scriptstyle`、`\scriptscriptstyle` に対応しています。

先ほどの `\frac`、`\tfrac`、`\binom` は `\genfrac` を使えば、次のように定義できます。

```
\newcommand\frac[2]{\genfrac{}{}{}{#1}{#2}}
\newcommand\tfrac[2]{\genfrac{}{}{1}{#1}{#2}}
\newcommand\binom[2]{\genfrac{()}{()}{0pt}{#1}{#2}}
```

連分数を表現する方法の一つとして `\cfrac` 命令を用いる事が考えられます。

```
\begin{displaymath}
\cfrac{1}{x+
\cfrac{1}{x+
\cfrac{1}{x+\dotsb}
}
}
\end{displaymath}
```

$$x + \frac{1}{x + \frac{1}{x + \dots}}$$

▼ 5.11.12 $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts の数学記号

ここで $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts の記号を出力するためにはプリアンブルで `amssymb` パッケージを読み込みます*7。

表 5.30 $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts の二項演算子

$\dot{+}$ <code>\dotplus</code>	\boxtimes <code>\boxtimes</code>	\curlywedge <code>\curlywedge</code>
\smallsetminus <code>\smallsetminus</code>	\boxdot <code>\boxdot</code>	\curlyvee <code>\curlyvee</code>
\cap <code>\Cap</code>	\boxplus <code>\boxplus</code>	\circleddash <code>\circleddash</code>
\cup <code>\Cup</code>	\div <code>\divideontimes</code>	\circledast <code>\circledast</code>
$\bar{\wedge}$ <code>\barwedge</code>	\ltimes <code>\ltimes</code>	\circledcirc <code>\circledcirc</code>
\veebar <code>\veebar</code>	\rtimes <code>\rtimes</code>	\cdot <code>\centerdot</code>
$\overline{\wedge}$ <code>\doublebarwedge</code>	\leftthreetimes <code>\leftthreetimes</code>	\intercal <code>\intercal</code>
\boxminus <code>\boxminus</code>	\rightthreetimes <code>\rightthreetimes</code>	

5.12 txfonts/pxfonts での拡張

Young Ryu 氏による txfonts/pxfonts では数学記号に関する拡張が行われています。これらの数学記号を出力する方法は第 5 章を参照してください。



何らかの事情により txfonts/pxfonts に含まれる特定の記号だけが必要になった場合は、例えば次のように `\usefont` と `\symbol` 命令を使う事で**その場しのぎ的に**用いる事ができます。

```
\newcommand*{\myTxsyc[1]{\text{%
  \usefont{U}{txsyc}{m}{n}%
  \symbol{#1}}}}
\newcommand*\multiMapDotBothA
  {\myTxsyc{"17}}
\newcommand*\circledDotLeft
  {\mathrel{\myTxsyc{"93}}}%
\begin{eqnarray*}
x \multiMapDotBothA y & \& \neq & \& x \\
\mathrel{\multiMapDotBothA} y & \& \& \& \\
x \circledDotLeft y & \& \& \& \\
\end{eqnarray*}
```

$$x \circ \bullet y \neq x \circ \bullet y$$

$$x \circ \rightarrow y$$

*7 amssymb パッケージを読み込むと自動的に amsfonts パッケージも読み込まれます。

表 5.31 $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts の二項関係子

\leq	<code>\leqq</code>	\approx	<code>\preccurlyeq</code>	\sim	<code>\thicksim</code>
\leq	<code>\leqslant</code>	\triangleleft	<code>\vartriangleleft</code>	\approx	<code>\thickapprox</code>
\leq	<code>\eqslantless</code>	\triangleleft	<code>\trianglelefteq</code>	\supseteq	<code>\supseteqq</code>
\lesssim	<code>\lesssim</code>	\dashv	<code>\vDash</code>	\supset	<code>\Supset</code>
\lessapprox	<code>\lessapprox</code>	\Vdash	<code>\Vdash</code>	\sqsupseteq	<code>\sqsupseteq</code>
\approx	<code>\approxeq</code>	\smile	<code>\smallsmile</code>	\succcurlyeq	<code>\succcurlyeqq</code>
\lessdot	<code>\lessdot</code>	\frown	<code>\smallfrown</code>	\curlyeqsucc	<code>\curlyeqsucc</code>
\lll	<code>\lll</code>	\bumpeq	<code>\bumpeq</code>	\succsim	<code>\succsim</code>
\lessgtr	<code>\lessgtr</code>	\Bumpeq	<code>\Bumpeq</code>	\succapprox	<code>\succapprox</code>
\lesseqgtr	<code>\lesseqgtr</code>	\geqq	<code>\geqq</code>	\vartriangleright	<code>\vartriangleright</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\geqslant	<code>\geqslant</code>	\trianglerighteq	<code>\trianglerighteq</code>
\doteqdot	<code>\doteqdot</code>	\geqslantgtr	<code>\geqslantgtr</code>	\Vdash	<code>\Vdash</code>
\risingdotseq	<code>\risingdotseq</code>	\gtrsim	<code>\gtrsim</code>	\shortmid	<code>\shortmid</code>
\fallingdotseq	<code>\fallingdotseq</code>	\gtrapprox	<code>\gtrapprox</code>	\shortparallel	<code>\shortparallel</code>
\backsim	<code>\backsim</code>	\gtrdot	<code>\gtrdot</code>	\between	<code>\between</code>
\backsimeq	<code>\backsimeq</code>	\ggg	<code>\ggg</code>	\pitchfork	<code>\pitchfork</code>
\subseteq	<code>\subseteq</code>	\gtrless	<code>\gtrless</code>	\varpropto	<code>\varpropto</code>
\Subset	<code>\Subset</code>	\gtreqless	<code>\gtreqless</code>	\blacktriangleleft	<code>\blacktriangleleft</code>
\sqsubset	<code>\sqsubset</code>	\gtreqqless	<code>\gtreqqless</code>	\therefore	<code>\therefore</code>
\preccurlyeq	<code>\preccurlyeq</code>	\eqcirc	<code>\eqcirc</code>	\backepsilon	<code>\backepsilon</code>
\curlyeqprec	<code>\curlyeqprec</code>	\circeq	<code>\circeq</code>	\blacktriangleright	<code>\blacktriangleright</code>
\precsim	<code>\precsim</code>	\trianglelefteq	<code>\trianglelefteq</code>	\because	<code>\because</code>

`\circledDotLeft` の方は `\mathrel` を明示的に指定しているため、適切な関係子の空白が挿入されていますが、`\multiMapDotBothA` の方は空きが適切ではありません。一部分だけ関係子として使うような場合には、`\mathrel` を直接記述します。

こうすると、もし本文で（標準の）Computer Modern フォントを使っている場合、複数のファミリーが混在する事になりますので、積極的に推奨される方法とは言えません。

表 5.32 $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts の否定二項関係子

\nless	\ntriangleleft	\nsucceq
\nleq	\ntrianglelefteq	\succsim
\nleqslant	\nsubseteq	\succapprox
\nleqq	\subsetneq	\ncong
\lneq	\varsubsetneq	\nshortparallel
\lneqq	\subsetneqq	\nparallel
\lvertneqq	\varsubsetneqq	\nvDash
\lnsim	\ngtr	\nVDash
\lnapprox	\ngeq	\ntriangleright
\nprec	\ngeqslant	\ntrianglerighteq
\npreceq	\ngeqq	\nsupseteq
\precnsim	\gneq	\nsupseteqq
\precnapprox	\gneqq	\supseteq
\nsim	\gvertneqq	\varsupseteq
\nshortmid	\gnsim	\supseteqq
\nmid	\gnapprox	\varsupseteqq
\nvDash	\nsucc	
\nVDash	\nsucceq	

表 5.33 $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts の矢印記号

\dashrightarrow	\Lsh	\rightarrowtail
\dashleftarrow	\uparrows	\looparrowright
\leftleftarrows	\upharpoonleft	\rightleftharpoons
\leftrightarrows	\downharpoonleft	\curvearrowright
\Lleftarrow	\multimap	\circlearrowright
\twoheadleftarrow	\leftrightsquigarrow	\Rsh
\leftarrowtail	\rightrightarrows	\downdownarrows
\looparrowleft	\rightleftarrows	\upharpoonright
\leftrightharpoons	\rightrightarrows	\downharpoonright
\curvearrowleft	\rightleftarrows	\rightsquigarrow
\circlearrowleft	\twoheadrightarrow	

表 5.34 $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts の否定矢印記号

\nleftarrow	\nleftrightarrow	\nLeftarrow
\nrightarrow	\nLeftrightarrow	

表 5.35 $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts のギリシャ文字とヘブライ文字

\daleth `\digamma` \beth `\beth` \gimel `\gimel`
 \varkappa `\varkappa` \daleth `\daleth`

表 5.36 $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts の区切り記号

\ulcorner `\ulcorner` \urcorner `\urcorner` \llcorner `\llcorner` \lrcorner `\lrcorner`

表 5.37 その他の $\mathcal{A}\mathcal{M}\mathcal{S}$ Fonts 数式記号

\hbar <code>\hbar</code>	\nexists <code>\nexists</code>	\blacksquare <code>\blacksquare</code>
\hslash <code>\hslash</code>	\mho <code>\mho</code>	\blacklozenge <code>\blacklozenge</code>
\vartriangle <code>\vartriangle</code>	\Finv <code>\Finv</code>	\bigstar <code>\bigstar</code>
∇ <code>\nabla</code>	\Game <code>\Game</code>	\sphericalangle <code>\sphericalangle</code>
\square <code>\square</code>	\Bbbk <code>\Bbbk</code>	\complement <code>\complement</code>
\lozenge <code>\lozenge</code>	\backprime <code>\backprime</code>	\eth <code>\eth</code>
\textcircled{S} <code>\textcircled{S}</code>	\varnothing <code>\varnothing</code>	\diagup <code>\diagup</code>
\angle <code>\angle</code>	\blacktriangle <code>\blacktriangle</code>	\diagdown <code>\diagdown</code>
\measuredangle <code>\measuredangle</code>	\blacktriangledown <code>\blacktriangledown</code>	

表 5.38 その他の文字記号

\checkmark `\checkmark` \textcircled{R} `\textcircled{R}` \maltese `\maltese` \yen `\yen`

表 5.39 $\text{txfonts}/\text{pxfonts}$ で拡張された二項演算子

\medcirc <code>\medcirc</code>	\nplus <code>\nplus</code>	\sqcapplus <code>\sqcapplus</code>
\medbullet <code>\medbullet</code>	\boxast <code>\boxast</code>	\rhd <code>\rhd</code>
\invamp <code>\invamp</code>	\boxbslash <code>\boxbslash</code>	\lhd <code>\lhd</code>
\circledwedge <code>\circledwedge</code>	\boxbar <code>\boxbar</code>	\unrhd <code>\unrhd</code>
\circledvee <code>\circledvee</code>	\boxslash <code>\boxslash</code>	\unlhd <code>\unlhd</code>
\circledbar <code>\circledbar</code>	\Wr <code>\Wr</code>	
\circledbslash <code>\circledbslash</code>	\sqcupplus <code>\sqcupplus</code>	

表 5.40 txfonts/pxfonts で拡張された数学記号

α \alphaup	ν \nuup	ω \omegaup
β \betaaup	ξ \xiup	\diamond \Diamond
γ \gammaaup	π \piup	\diamond \Diamonddot
δ \deltaaup	ϖ \varpiup	\blacklozenge \Diamondblack
ϵ \epsilonup	ρ \rhoup	λ \lambdaslash
ε \varepsilonup	ϱ \varrhoup	λ \lambdabar
ζ \zetaaup	σ \sigmaup	\clubsuit \clubsuit
η \etaaup	ς \varsigmaup	\blacklozenge \vardiamondsuit
θ \thetaaup	τ \tauup	\heartsuit \varheartsuit
ϑ \varthetaaup	υ \upsilonup	\spadesuit \varspadesuit
ι \iotaaup	ϕ \phiup	\top \Top
κ \kappaaup	φ \varphiup	\bot \Bot
λ \lambdaaup	χ \chiup	
μ \muup	ψ \psiup	

表 5.41 txfonts/pxfonts で拡張された大型演算子

\oplus \bignplus	\int \sqint	\oint \oiintctrlockwise
\oplus \bigsqcupplus	\int \sqiintop	\oint \oiintclockwise
\oplus \bigsqcapplus	\int \sqiiintop	\oint \varoiintctrlockwise
\sqcap \bigsqcap	\int \fint	\oint \varoiintclockwise
\sqcap \bigsqcap	\int \iint	\oint \oiiintctrlockwise
\times \varprod	\int \iiint	\oint \oiiintclockwise
\oint \oiint	\int \iiiint	\oint \varoiiiintctrlockwise
\oint \oiiint	\int \dotsint	\oint \varoiiintclockwise
\oint \ointctrlockwise		\oint \ointclockwise
\oint \varointctrlockwise		\oint \varointclockwise

表 5.42 txfonts/pxfonts で拡張された区切り記号

\llbracket \llbracket \rrbracket \rrbracket \lrcorner \lrcorner \rbracket \rbracket

表 5.43 txfonts/pxfonts での変体文字

g \varg v \varv w \varw y \vary

表 5.44 txfonts/pxfonts で拡張された二項関係子

\leftarrow	<code>\mappedfrom</code>	$\not\leq$	<code>\ngtrless</code>	\Join	<code>\Join</code>
\longleftarrow	<code>\longmappedfrom</code>	$\not\lessgtr$	<code>\nlessgtr</code>	\times	<code>\openJoin</code>
\mapsto	<code>\Mapsto</code>	\neq	<code>\nbumpeq</code>	\rtimes	<code>\lrtimes</code>
\Longrightarrow	<code>\Longmapsto</code>	\neq	<code>\nBumpeq</code>	\times	<code>\opentimes</code>
\leftarrow	<code>\Mappedfrom</code>	\backsimeq	<code>\backsim</code>	\nsqsubset	<code>\nsqsubset</code>
\longleftarrow	<code>\Longmappedfrom</code>	\backsimeq	<code>\backssimeq</code>	\nsqsupset	<code>\nsqsupset</code>
\mapsto	<code>\mmapsto</code>	\neq	<code>\ne</code>	\dashleftarrow	<code>\dashleftarrow</code>
\Longrightarrow	<code>\longmmapsto</code>	\asymp	<code>\nasymp</code>	\dashrightarrow	<code>\dashrightarrow</code>
\leftarrow	<code>\mmappedfrom</code>	\equiv	<code>\nequiv</code>	\leftrightarrow	<code>\dashleftarrow</code>
\longleftarrow	<code>\longmmappedfrom</code>	\sim	<code>\nsim</code>	\curvearrowleft	<code>\leftsquigarrow</code>
\mapsto	<code>\Mmapsto</code>	\approx	<code>\napprox</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>
\Longrightarrow	<code>\Longmmapsto</code>	\subset	<code>\nsubset</code>	\twoheadleftarrow	<code>\twoheadleftarrow</code>
\leftarrow	<code>\Mmappedfrom</code>	\supset	<code>\nsupset</code>	\nearrow	<code>\Nearrow</code>
\longleftarrow	<code>\Longmmappedfrom</code>	\nll	<code>\nll</code>	\searrow	<code>\Searrow</code>
$//$	<code>\varparallel</code>	\ngg	<code>\ngg</code>	\nwarrow	<code>\Nwarrow</code>
\parallel	<code>\varparallelinv</code>	\thickapprox	<code>\nthickapprox</code>	\swarrow	<code>\Swarrow</code>
$\#$	<code>\nvarparallel</code>	\approx	<code>\napprox</code>	\perp	<code>\Perp</code>
$\#$	<code>\nvarparallelinv</code>	\prec	<code>\nprec</code>	\leadsto	<code>\leadsto</code>
\approx	<code>\colonapprox</code>	\succ	<code>\nsucc</code>	\leadsto	<code>\leadsto</code>
\sim	<code>\colonsim</code>	\preceq	<code>\npreceq</code>	\square	<code>\boxright</code>
\approx	<code>\Colonapprox</code>	\succeq	<code>\nsucceq</code>	\square	<code>\boxleft</code>
\sim	<code>\Colonsim</code>	\simeq	<code>\nsimeq</code>	\square	<code>\boxdotright</code>
\doteq	<code>\doteq</code>	\notin	<code>\notin</code>	\square	<code>\boxdotleft</code>
\circ	<code>\multimapinv</code>	\notni	<code>\notni</code>	\diamond	<code>\Diamondright</code>
\circ	<code>\multimapboth</code>	\Subset	<code>\nSubset</code>	\diamond	<code>\Diamondleft</code>
\bullet	<code>\multimapdot</code>	\Supset	<code>\nSupset</code>	\diamond	<code>\Diamonddotright</code>
\bullet	<code>\multimapdotinv</code>	\sqsubset	<code>\nsqsubset</code>	\diamond	<code>\Diamonddotleft</code>
\bullet	<code>\multimapdotboth</code>	\sqsupset	<code>\nsqsupset</code>	\square	<code>\boxRight</code>
\bullet	<code>\multimapdotbothA</code>	\coloneqq	<code>\coloneqq</code>	\square	<code>\boxLeft</code>
\bullet	<code>\multimapdotbothB</code>	\eqqcolon	<code>\eqqcolon</code>	\square	<code>\boxdotRight</code>
\Vdash	<code>\VDash</code>	\coloneq	<code>\coloneq</code>	\square	<code>\boxdotLeft</code>
\VvDash	<code>\VvDash</code>	\eqcolon	<code>\eqcolon</code>	\diamond	<code>\DiamondRight</code>
\cong	<code>\cong</code>	\Coloneqq	<code>\Coloneqq</code>	\diamond	<code>\DiamondLeft</code>
\preceq	<code>\preceq</code>	\Eqqcolon	<code>\Eqqcolon</code>	\diamond	<code>\DiamonddotRight</code>
\succeq	<code>\succeq</code>	\Coloneq	<code>\Coloneq</code>	\diamond	<code>\DiamonddotLeft</code>
\prec	<code>\nprec</code>	\Eqcolon	<code>\Eqcolon</code>	\circ	<code>\circleft</code>
\succ	<code>\nsucc</code>	\strictif	<code>\strictif</code>	\circ	<code>\circleft</code>
\less	<code>\nless</code>	\strictfi	<code>\strictfi</code>	\circ	<code>\circdotright</code>
\ngtr	<code>\ngtr</code>	\strictiff	<code>\strictiff</code>	\circ	<code>\circdotleft</code>
\less	<code>\nless</code>	\circledless	<code>\circledless</code>	\circ	<code>\multimapbothvert</code>
\ngtr	<code>\ngtr</code>	\circledgtr	<code>\circledgtr</code>	\bullet	<code>\multimapdotbothvert</code>
\preccurlyeq	<code>\npreccurlyeq</code>	\lJoin	<code>\lJoin</code>	\bullet	<code>\multimapdotbothAvert</code>
\succcurlyeq	<code>\nsucccurlyeq</code>	\rJoin	<code>\rJoin</code>	\bullet	<code>\multimapdotbothBvert</code>

第 6 章

図表の構成

レポート・論文に図や表を取り入れる事は読者の理解を助ける事になります。この章では文書中にどのように図表を構成すれば良いのかを解説します。

6.1 一般的な取り決め

以下は一般的なレポート・論文作成における図表に関する取り決めです。

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事がなければ**、本文中に配置する事も可能です。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事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事もありますが、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**、図表の下部に配置します。

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**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事になります。

これらの取り決めを守る事により図表に関する一貫性がうまれる事となります。必ず上記のようにしなければならないというルールはありません。学会や機関によってはこれとは異なる方針を持っている事もあります。いずれにしても、何かしらの一貫性を持たせるという基本原則は守るように執筆する事を心がけるようにすると、読者に親切な表記と言えるでしょう。

6.2 L^AT_EX での扱い

レポートや論文においては基本的に図表に対して通し番号を振るために、図表は `table` 環境か `figure` 環境に入れ子にします。この場合、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 6.1)。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

表 6.1 浮動体の種類

	表	図
入れる環境	<code>table</code> 環境	<code>figure</code> 環境
見出しの位置	表の上部	図の下部

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。ユーザはこれら図表 (浮動体) の配置場所を指定する事ができます。指定できる場所は表 6.2 となります。位置指定は複数指定する事が可能です。

これらの位置指定は `table` 環境や `figure` 環境の**任意引数として渡します**。`figure` 環境で例を示すと次のように使います。

```
\begin{figure}[htbp]
```

表 6.2 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

ここに図が入ります。
`\end{figure}`

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事ができます。他にも環境中に文字列を挿入する事も可能です。

一般的なレポート・論文等においては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事になります。間違っても**手動で図表の番号を書かないで下さい**。

6.3 表

\LaTeX で表を作るために三つの環境が用意されています。

tabbing 環境 タブを制御する事によって表を作成する。

tabular 環境 高度な表も作成する事ができる汎用的な表作成環境。

array 環境 `tabular` 環境と機能は類似しているが数式の行列などに使われる事が多い。

array 環境は 5.5.5 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>}
  要素11 & ... & 要素1n \\
  ⋮ & ⋮ & ⋮ \\
  要素m1 & ... & 要素mn \\
\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事です。

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 環境で使用できる主な列指定子は表 6.3 の通りです。tabular 環境に

表 6.3 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>}{<列指定子>}	回数分だけ<列指定子>を繰り返す。

における各要素（成分）はアンパサンド‘&’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}
\LaTeX\,2.09 & \LaTeXe & \LaTeX\,3 \\
\end{tabular}
```

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 6.4）。

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 6.4 tabular 環境中での罫線の命令

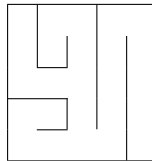
命令	意味
<code>\hline</code>	横に引けるだけの罫線を引きます
<code>\hline\hline</code>	引けるだけの 2 重の横罫線を引きます
<code>\vline</code>	要素の中で引けるだけの縦罫線を引きます
<code>\cline{<範囲>}</code>	要素の罫線を行の範囲を指定して引きます
<code>\multicolumn{<数値>}{<要素>}{<列指定子>}</code>	行をつなげて列指定子通りに要素を出力します

```
\begin{tabular}{|c|c|c|}
\hline\hline
\multicolumn{3}{|c|}{{\LaTeX}}\
\hline
\LaTeX\,2.09 & \LaTeXe & \LaTeX\,3\
\cline{2-3}
\end{tabular}
```

L ^A T _E X		
L ^A T _E X 2.09	L ^A T _E X 2 _ε	L ^A T _E X 3

罫線を利用して迷路のようなものも作れます。

```
\begin{tabular}{|ccc|c|c|}
\hline
& \multicolumn{1}{|c|}{ } & & \\
& \multicolumn{1}{c}{ } & & \\
& \multicolumn{1}{|c|}{ } & & \\
\cline{2-2}
& & & \\
& & & \\
& \multicolumn{1}{c}{ } & & \\
\cline{2-2}
& & \multicolumn{1}{c}{ } & & \\
& & \multicolumn{1}{c}{ } & & \\
\hline
\end{tabular}
```



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るので以下のようなフォーマットになります。

```
\begin{table}[htpb]
\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& \LaTeX\,2.09 & {\LaTeXe}& \LaTeX\,3 \\ \hline
\end{tabular}
\end{center}
\end{table}

```

上記のソースの出力例が表 6.5 となります。

表 6.5 表の出力例

出力例	1	2	3
\LaTeX の遷移	\LaTeX 2.09	$\LaTeX 2_{\epsilon}$	\LaTeX 3

▷ **例題 6.1** しかし、さすがに毎回同じような記述をしていたのでは疲れますので、自前で表用の `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}}

```

このように定義すれば、次のように簡単に用いる事ができるようになります。実際に上記の定義を用いてタイプセットし、実行結果を吟味してください。

```

\begin{mytab}[htbp]{中央揃えで見出しのある表の環境}{tab:hoge}
\begin{tabular}[t]{c}
\LaTeX\,2.09 & \LaTeXe & \LaTeX\,3\\
\end{tabular}
\end{mytab}

```

▶ **問題 6.2** 問題 3.3 では `\maketitle` という表題を出力するための命令を紹介しました。そこでは `\and` によって著者を列挙する事ができました。それと等価な出力を `tabular` 環境で実装できるかどうか考えて下さい。

おおむね次のような方法でも実装できる事を確認してください。

```

\newcommand \AND{\end{tabular}\hspace{1zw}\begin{tabular}[t]{c}}
\newcommand \makeAUTHOR[1]{%
\begin{center}\begin{tabular}[t]{c}#1\end{tabular}\end{center}}
\makeAUTHOR{夏目漱石 \\ ○○研究所 \\ ○○事業部 \AND
福澤諭吉 \\ △△株式会社 \\ △△研究所\AND
芥川龍之介 \\ □□大学 □□学部 \\ □□学科}

```

▼ 6.3.1 表中の脚注

tabular 環境中での脚注はうまく出力できない事が多いようです。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]{一つ目の脚注です。}
\footnotetext[2]{二つ目の脚注です。}
\footnotetext[3]{三つ目の脚注です。}
\\ ちよつと表示が変になっています。
```

一つ目 ^{*1}	二つ目 ^{*2}	三つ目 ^{*3}
-------------------	-------------------	-------------------

ちよつと表示が変になっています。

-
- ^a 一つ目の脚注です。
 - ^b 二つ目の脚注です。
 - ^c 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事もできます。

```
\begin{tabular}{|c|c|c|} \hline
  一つ目${}^{\{a\}}$ & 二つ目${}^{\{b\}}$ &
  三つ目${}^{\{c\}}$ \\ \hline
\end{tabular}
{\footnotesize \\
${}^{\{a\}}$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。}
```

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。

▼ 6.3.2 tabular 環境の出力位置

実は tabular 環境は列指定子の前に任意引数を一つとります。

```
\begin{tabular}[<位置指定子>]{<列指定子>}
<表を構成するための記述>
\end{tabular}
```

これは表の位置と段落の位置を調整するものです。tabular 環境で作成された表の上部と段落の位置を合わせるときは 't' を、下部ならば 'b' を、中央ならば 'c' を選びます。

```
\newcommand{\testtab}[1][c]{~日本~
\begin{tabular}[#1]{|c|} \hline
函館\\ 未来\\ \hline\end{tabular}}
\testtab \testtab[t] \testtab[b]
```

函館 未来	日本	函館 未来	日本	函館 未来	日本	函館 未来
----------	----	----------	----	----------	----	----------

▼ 6.3.3 書籍スタイルの表罫線——booktabs

日本人のスタイルの慣習として表組みで縦罫線や斜線を使う傾向が見られるようです。典型的な (typical) 日本人が組んだものは下記のようになります。

```
\begin{tabular}{|l|l|l|}
\hline
名称 & 型番 & 個数 \\
\hline\hline
たわし & TWS01 & 1000 \\
\hline
石鹸 & SP01 & 5000 \\
\hline
\end{tabular}
```

名称	型番	個数
たわし	TWS01	1000
石鹸	SP01	5000

実際の本作りや欧文での表組みでは上記のような組み方は避けた方が無難です。認知心理学的にもやさしい次のような組み方をお勧めします。

```
\begin{tabular}{l|l|l}
\hline
名称 & 型番 & 個数 \\
\hline
たわし & TWS01 & 1000 \\
\hline
石鹸 & SP01 & 5000 \\
\hline
\end{tabular}
```

名称	型番	個数
たわし	TWS01	1000
石鹸	SP01	5000

ただ、もう少し本格的にやろうと思えば、Simon Fear 氏による booktabs パッケージを使うと良いでしょう。こちらの方が書籍に近いスタイルとなります。

<code>\toprule</code>	(表の最上部に引く)	<code>\midrule</code>	(表の中間に引く)
<code>\bottomrule</code>	(表の最下部に引く)	<code>\cmidrule</code>	{(罫線を引く範囲)}

`\toprule` と `\midrule`、そして `\bottomrule` の三つを必ず使うようにします。


```
\usepackage{booktabs}
\begin{tabular}{l}
\toprule
品名 & 番号 & 個数 \\
\midrule
たわし & 02A & 3 \\
雑巾 & 55B & 2 \\
傘 & X2B & 5 \\
\bottomrule
\end{tabular}
```

品名	番号	個数
たわし	02A	3
雑巾	55B	2
傘	X2B	5

表の中に半端の罫線を引く場合は `\cmidrule` 命令を使います。 `\cmidrule` は `\multicolumn` などにより列を連結した場合等に使う事ができると思います。

```
\usepackage{booktabs}
\begin{tabular}{l}
\toprule
\multicolumn{2}{c}{項目} \\
\cmidrule{1-2}
品名 & 型番 & 個数 \\
\midrule
たわし & 02A & 3 \\
雑巾 & 55B & 2 \\
傘 & X2B & 5 \\
\bottomrule
\end{tabular}
```

項目		
品名	型番	個数
たわし	02A	3
雑巾	55B	2
傘	X2B	5

▼ 6.3.4 小数点揃え——`dcolumn`

`tabular` 環境などで表を作っていると、小数点などで列を整列させたいときがあります。この場合、手動で次のようにもできます。

```
\begin{center}
\begin{tabular}{l}
 $\sqrt{157}$  & 12 & 53 \\
 $\pi$  & 3 & 141592
\end{tabular}
\end{center}
```

しかし、ここは自動的に小数点でそろえて欲しいものです。小数点などをそろえる一つの方法として David Carlisle 氏の `dcolumn` を使う方法があります。

`D`{ \TeX での区切り}{ DVI での出力形式}{小数部の桁数}
`\newcolumnntype`{区切り記号}{入出力に関する設定}

と定義する事により、小数点‘.’に限らず、なんらかの区切りで列を整列できます。

```
\usepackage{dcolumn}
\begin{center}
\newcolumntype{.}{D{.}{.}{6}}
\begin{tabular}{|l|.l|}

$$\begin{array}{|l|.l|}
\sqrt{157} & 12.53 \\
\pi & 3.141592
\end{array}$$

\end{tabular}
\end{center}
```

`tabular` 環境などで直接列指定子‘D’を使う事もできます。上記の場合はあらかじめピリオド‘.’を列の整列用の指定子として登録しています。

▼ 6.3.5 表における行の連結——`multirow`

`array/tabular` 環境で表などを作成していると、列の連結を行なう事がしばしばあります。

```
\begin{tabular}{|l|l|}
\multicolumn{2}{c}{中央} & 右側 \\
左側 & 中央 & 右側 \\
\end{tabular}
```

しかし、行の連結となると結構面倒です。そこで Jerry Jeichter 氏と Piet Oostrum 氏による `multirow` パッケージを使えば良いでしょう。

```
\multirow{<行数>}{<幅>}{<要素>}
\multirow{<行数>}*{<要素>}
```

星を付けた場合は<要素>を LR モードで組んだときの幅で表を配置します。まずは行を連結しない場合です。

```
\usepackage{multirow}
\begin{tabular}{|l|l|l|}
\hline
\multicolumn{2}{|c|}{新商品} & 旧商品 \\
なべ & やかん & たわし \\
\hline
\end{tabular}
```

新商品		旧商品
なべ	やかん	たわし

次は行を<要素>分の幅で連結した場合です。

```
\begin{tabular}{|l|l|}
\hline
\multirow{2}{*}{新商品}
& なべ\\
& やかん\\ \hline
旧商品 & たわし\\ \hline
\end{tabular}
```

新商品	なべ やかん
旧商品	たわし

最後に全角 1 文字分の幅で行を四つ連結させた例です。ただし、最後の行が 3 文字分あるため、幅の指定は効力がありません。

```
\begin{tabular}{|c|l|}
\hline
\multirow{4}{1zw}{新商品}
& なべ \\
& やかん \\
& コップ \\
& 洗剤 \\ \hline
旧商品 & たわし \\ \hline
\end{tabular}
```

新 商 品	なべ やかん コップ 洗剤
	たわし
旧商品	たわし

▼ 6.3.6 ページを跨ぐ表——longtable

要素が非常に多い時は、表がページに収まりきらない場合があります。このような場合は David Carlisle 氏の longtable パッケージでも使いましょう。longtable パッケージを読み込む事により longtable 環境が使えるようになります。

ただし、table 環境中にはいれません。また表の幅をそろえるためには、longtable パッケージの警告が出なくなるまで複数回のタイプセットが必要になります。

ページが複数ページに跨いでしまったときに、各ページの下部・上部に表示させたい要素が指定できます。

<要素> \endfirsthead	(表の最初のページの上部にだけ表示する行の要素)
<要素> \endhead	(行がページを跨ぐときページ上部に表示する要素)
<要素> \endfoot	(行がページを跨ぐときページ下部に表示する要素)
<要素> \endlastfoot	(表の最後のページの下部だけに表示する行の要素)

具体例を見た方が分かりやすいでしょう。次のような入力があるとするとき出力は図 6.1 となります。

```

\documentclass[a4j,11pt,papersize]{jsarticle}
\usepackage{longtable}
\newcommand\hoge[1][0]{%
  醤油 #1-0 & 32892378923894832894 & 1000 \\
  醤油 #1-1 & 32892378923894832894 & 1000 \\
  醤油 #1-2 & 32892378923894832894 & 1000 \\
  醤油 #1-3 & 32892378923894832894 & 1000 \\
  醤油 #1-4 & 32892378923894832894 & 1000 \\
  醤油 #1-5 & 32892378923894832894 & 1000 \\
  醤油 #1-6 & 32892378923894832894 & 1000 \\
  醤油 #1-7 & 32892378923894832894 & 1000 \\
  醤油 #1-8 & 32892378923894832894 & 1000 \\
  醤油 #1-9 & 32892378923894832894 & 1000 \\
}
\begin{document}
% 表の幅を取得するために \jobname.aux に longtable パッケージは
% 情報を書き出し、2 回目以上のタイプセットで幅をそろえる。
\newcommand\mytablehead{\hline 商品 & 番号 & 個数 \\}
\begin{longtable}{|l|l|l|}
\caption{長いながーい表\label{longtable}}
% 表の最初のページの上部にだけ表示する行の要素
\endfirsthead
\hline
\multicolumn{3}{|c|}{前ページの表の続きです。}\\
\mytablehead
\hline
% 行がページを跨ぐとき、各ページの上部に表示する行の要素
\endhead
\hline
\multicolumn{3}{|c|}{この表の続きが次ページにあります。}\\
\hline
% 行がページを跨ぐとき、各ページの下部に表示する行の要素
\endfoot
\multicolumn{3}{|c|}{これでこの表は終わりです。}\\
\hline
% 表の最後のページの下部だけに表示する行の要素
\endlastfoot
% 実際の表の始まり
\mytablehead
\hline
\hoge[1] \hoge[2] \hoge[3] \hoge[4] \hoge[5]
\hoge[6] \hoge[7] \hoge[8] \hoge[9] \hoge[10]
\hoge[11] \hoge[12] \hoge[13]
\hline
\end{longtable}
\end{document}

```


▼ 6.3.7 表の幅の指定——tabularx

LaTeX 2_ε の array/tabular 環境では列の幅を直接指定できる列指定子 ‘p{<幅>}’ が用意されていますが、原稿を執筆している段階ではその幅を決定できない事がしばしばあります。自動的にその幅を求めてくれるような環境があれば便利です。そこで David Carlisle 氏の作成した tabularx パッケージを用いる事で tabularx 環境が使えます。

```
\begin{tabularx}{<幅>}{<列指定子>}
<表を構成する要素>
\end{tabularx}
```

具体例を以下に示します。

```
\documentclass[a4j,10pt,papersize]{jsarticle}
\usepackage{tabularx}
\makeatletter
\def\hoge{\@tempcnta=\z@ \@whilenum \@tempcnta<10\do{%
  ○○○○\advance\@tempcnta\@ne}。}
\makeatother
\begin{document}
\hoge%
\begin{center}
\begin{tabularx}{\linewidth}{|X|X|X|}
\hline
\hoge & \hoge & \hoge \\
\hline
\end{tabularx}
\end{center}
\hoge%
\begin{center}
\begin{tabularx}{\linewidth}{|r|l|X|l|X|}
\hline
商品 & 値段 & 説明 & 型番 & 補足事項 \\
\hline
鍋 & 500 & \hoge & 59A & \hoge \\
\hline
やかん & 300 & \hoge & 9JA & \hoge \\
\hline
\end{tabularx}
\end{center}
\hoge%
\end{document}
```

上記の入力の出力例は図 6.2 となります。

○○○○○○○○○○	○○○○○○○○○○	○○○○○○○○○○
○○○○○○○○○○	○○○○○○○○○○	○○○○○○○○○○
○○○○○○○○○○	○○○○○○○○○○	○○○○○○○○○○
○○○○○○○○○○。	○○○○○○○○○○。	○○○○○○○○○○。

商品	値段	説明	型番	補足事項
鍋	500	○○○○○○○○○○ ○○○○○○○○○○ ○○○○○○○○○○ ○○○○○○○○。	59A	○○○○○○○○○○ ○○○○○○○○○○ ○○○○○○○○○○ ○○○○○○○○。
やかん	300	○○○○○○○○○○ ○○○○○○○○○○ ○○○○○○○○○○ ○○○○○○○○。	9JA	○○○○○○○○○○ ○○○○○○○○○○ ○○○○○○○○○○ ○○○○○○○○。

図 6.2 tabularx 使用例の出力結果

tabularx 環境において列指定子 ‘X’ が新たに使えるようになっています。tabularx 環境は組むべき表の幅を知る必要があります。‘X’ が複数の場合はそれぞれの列の幅は均等な長さになります。

▼ 6.3.8 表作成支援ツール

L^AT_EX でゼロ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浦壁厚郎氏の Excel2tabular^{*1}などがありますので参考してください。これらのプログラムは Microsoft の Excel で作成された表を L^AT_EX 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 Calc2L^AT_EX^{*2}というものもあります。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 L^AT_EX に貼り付ける事ができます。

*1 <http://www.ne.jp/asahi/i/love/E2T/>

*2 <http://web.hc.keio.ac.jp/~mr041754/calc2latex/indexj.html>

最近では直接 \LaTeX から Excel ファイルを読み込める Hans-Peter Doerr 氏による `exceltex` パッケージがあります*3。Perl スクリプトを仲介する事で指定したセルやシートを読み込む事ができます。

6.4 図に関する制約と画像の扱い

図の挿入に関しては大きく分けて 2 通りの方法があります。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。

\LaTeX には `picture` 環境と呼ばれる簡単な作図をするための描画環境が用意されています。この `picture` 環境を拡張した `epic`, `eepic`, `pict2e` などが存在し、ある程度の作図ができるコマンドが用意されています。 `picture` 環境とその周辺の詳しい事は『 \LaTeX コンパニオン』[34] や『 \LaTeX グラフィックスコンパニオン』[35] を参照してください。

何らかの外部プログラムで作成した BMP, JPEG, PNG, EPS, PDF 等の画像を \LaTeX に張り込むためには、一般的には `graphicx` パッケージを用います。

\LaTeX 自身では画像ファイルを直接的に扱う仕組みは用意されておらず、画像ファイルに関する多くの処理をデバイスドライバという外部プログラムに依存した形を取るため、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ください。最終的に出力したい文書形式が PDF ならば `Dvipdfmx`, PostScript ならば `dvips` を使う事になります。近年では DVI 形式から直接 PDF を生成できる `Dvipdfmx` を使う事を強く推奨します。 `Dvipdfmx` を用いる事で BMP, JPEG, PNG, EPDF (単一ページの PDF), EPS 画像の張り込みが可能となり、さらに DVI ファイルから直接 PDF を生成する事ができます。

最近の動向として論文等の提出、印刷には PDF を用いる場合が増えているようです。 `Dvipdfmx` を使えば \LaTeX でそのまま PDF 画像の埋め込み等もサポートしているため、今後は何かしらの問題がない限り、 `Dvipdfmx` を使うようにすると何かと便利だと思われれます。

近年まで \LaTeX では EPS 以外の画像の張り込みは難しいという都市伝説的な定説がありましたが、現在は `Dvipdfmx` の登場により状況は幾分変化していますし、これからも変化すると考えられます。

*3 <http://www.ring.gr.jp/pub/text/CTAN/macros/latex/contrib/exceltex/>

6.5 画像ファイルの張り込み

LaTeX ではビットマップ画像や、曲線の描画などの多くの処理をデバイスドライバと呼ばれる外部のプログラムに依存しています。そのため、LaTeX 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事になります。

▼ 6.5.1 デバイスドライバの選択

各種のデバイスドライバプログラムにおける画像形式に対する対応状況を表 6.6 に示します（2006 年 4 月現在での対応状況）。星印がついているものは Ghostscript など

表 6.6 各種デバイスドライバの画像形式対応状況

デバイスドライバ	対応画像形式
xdvi	EPS*
dvips	EPS
Dvipdfmx	EPS*, EPDF, PNG, BMP, JPEG
dviout	EPS*, Susie plug-in により他の形式に対応可能

の外部プログラムを必要とする形式です。

LaTeX で画像を張り込む時、多くの場合は標準的に `graphicx` パッケージを使う事になります。Dvipdfmx を使っている場合はパッケージオプションを `dvipdfmx` とします。

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

これにより `graphicx` パッケージは `dvipdfmx.def` という設定ファイルを読み込みます。もしも `dvipdfmx.def` というファイルが存在しないようであれば、以下の URL からファイルを取得し `$texmf/tex/latex/graphics/` 等のディレクトリにコピーしてください*4。

古い TeX/LaTeX（2006 年以前）がインストールされているのであれば、`dvipdfmx` ではなく、`dvipdfm` オプションを指定して、Dvipdfmx で PDF をデバイスドライバとします*5。

*4 <http://tex.dante.jp/joul/dvipdfmx.def>

*5 何かしらの理由がない限り TeX 環境は定期的に更新する事が望ましいです。

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

Unix 系 OS ならば PostScript のほうが良いでしょうから *dvips* を *graphicx* パッケージのオプションとします。 *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、Windows であれば *dviout* も指定できます。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。 *dviout* ではプレビューも印刷も行えます。 *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^{*6}。

EPS 画像が多いならばいずれにしても 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れます。

▼ 6.5.2 具体的な手順

画像ファイルを L^AT_EX の文書に張り込むには、一般的に次のような手順を踏む事になります。

1. 外部プログラムで PDF や EPS 形式でファイルを保存。保存する時のオプションで可能であればフォントはアウトライン化し、カラーに依存しないようなファイルとします。
2. 文書のプリアンブルで *graphicx* パッケージを使う事を宣言します。
3. *graphicx* パッケージには**デバイスドライバ**を指定します。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 *dvipdfmx* を指定します。
4. EPS 以外の画像であれば L^AT_EX が解釈できる形で**バウンディングボックス**を指定します。
5. 図を挿入すべき場所に `\includegraphics` 命令を使ってファイル名を示します。

デバイスドライバの *Dvipdfmx* 等は画像ファイルを扱う事が可能ですが、L^AT_EX は画像ファイルを直接扱う事ができず、画像に関する情報を取得できません。そのため、

^{*6} *dviout* の場合 EPS 画像を取り込むときは Ghostscript にて EPS を PPM 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。

Dvipdfmx において JPEG, PNG, PDF, BMP 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）サイズ情報を与える事で張り込む事が可能です。一般的には画像の横の長さ縦の長さのサイズを指定する事となります。バウンディングボックスは〈filename〉.img という画像ファイルがあれば、〈filename〉.bb というファイルを graphicx パッケージが参照するようになっています。

Dvipdfm に付属する ebb というプログラムで画像のバウンディングボックス情報のファイル 〈filename〉.bb を作成できます。対応している画像形式は JPEG, PNG, PDF です*7。

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、ファイルの存在するディレクトリで

```
$ 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 を見てみます。

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

上記のように〈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、次のようにすれば〈filename〉.bb がなくても良い事になります。

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

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文字としたほうが互換性の上で安全です。

▷ **例題 6.3** 仮にファイル名が image.png の画像があったとすれば、コンソールから `ebb image.png` として image.png 用の image.bb が作成される事を確認してください。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、中身は次のようなものになっていると思います。

*7 ebb 以外にも identify コマンドや file コマンドでサイズ情報は知る事ができますし、Windows や Mac OS X であればエクスプローラやファインダーにも表示されます。

```
%%Title: ./image.png
%%Creator: ebb Version 0.5.2
%%BoundingBox: 0 0 595 841
```

‘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[papersize]{jsarticle}
\usepackage[dvipdfmx]{graphicx}
\begin{document}
\centering \includegraphics[width=4cm]{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 を作成します。これにより image.png が張り込まれた PDF が生成されるはずですが、

▼ 6.5.3 張り込みにおけるオプション

外部プログラムで作成して、すでに存在するような画像は `\includegraphics` 命令で張り込みます。

```
\includegraphics[〈設定〉]{〈ファイル名〉}
```

〈設定〉に関しては以下に示すようなオプションが使用できます。

- height=〈高さ〉 単位付きで画像の高さを指定します。
- totalheight=〈総合的な高さ〉 単位付きで画像の総合的な高さを指定します。
- 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。
- origin=〈原点〉 画像の基準点を決めます。
- bb=〈領域情報〉 **バウンディングボックス**と呼ばれる画像の大きさと原点座標を指定します。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。
- viewport=〈領域情報〉 画像の利用領域を指定します。切り抜きです。
- trim=〈領域情報〉 画像の端を切り抜きます。
- 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であろう領域が枠による代替表示になり、ファイル名を表示します。

`keepaspectratio` 拡大縮小したときに縦横比を保存するようにします。 `graphicx` パッケージの標準では保存されます。

▷ **例題 6.4** ご自分の持っている画像(ファイル)を(デバイス)が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 `gnu-head.pdf` と `gnu-head.bb` があると仮定します)。

```
\usepackage[dvipdfmx]{graphicx}
\includegraphics[width=3cm]
{images/gnu-head}
```



```
\usepackage[dvipdfmx]{graphicx}
\includegraphics[width=2cm,
trim=20 20 20 20]
{images/gnu-head}
```



```
\usepackage[dvipdfmx]{graphicx}
\includegraphics[width=2cm,
clip,viewport=131 304 459 548]
{images/gnu-head}
```



```
\usepackage[dvipdfmx]{graphicx}
\includegraphics[width=2cm,angle=30,
clip,viewport=131 304 459 548]
{images/gnu-head}
```



```
\usepackage[dvipdfmx]{graphicx}
\includegraphics[width=2cm,angle=90,
clip,viewport=131 304 459 548]
{images/gnu-head}
```



▼ 6.5.4 画像の拡大や回転等の操作

図などを反時計回りに 90° 回転させる事があるでしょう。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’ を使った事と同じです。`\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}
は
\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?

要素を**拡大縮小**するには `\scalebox` を使います。

```
\scalebox{⟨横の拡大率⟩}[⟨縦の拡大率⟩]{⟨要素⟩}
```

⟨拡大率⟩には長さを指定します。

```
\scalebox{2.3}{拡大縮小}\par
\scalebox{3}[1]{拡大縮小}
```

拡大縮小
拡大縮小

要素の**反転**には `\reflectbox` を使います。

```
\reflectbox{⟨要素⟩}
```

```
\reflectbox{文字列の反転}\par      罫又の𠄎宅文
\reflectbox{山は山}\par            山お山
\scalebox{-1}[1]{これも反転}      罫又よ𠄎こ
```

リサイズには `\resizebox` を使います。

```
\resizebox{<幅>}{<高さ>}{<要素>}
```

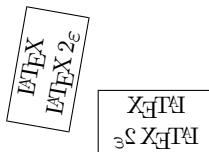
要素のリサイズ後の幅を<幅>に、高さを<高さ>にします。どちらか一方の拡大・縮小率に合わせたいときは‘!’を使います。

```
\resizebox{!}{1cm}{リサイズ}\par
\resizebox{3cm}{!}{リサイズ}
```

リサイズ リサイズ

以上の `\rotatebox`, `\scalebox`, `\reflectbox`, `\resizebox` は文字列、表、図、`minipage` 環境などの段落などにも使えます。

```
\newcommand{\testtab}{%
\begin{tabular}{|c|}
\hline \LaTeX\ \ \LaTeXe \ \ \hline
\end{tabular}}
\rotatebox{80}{\testtab}~
\reflectbox{\testtab}
```



▼ 6.5.5 Dvipsd mf x における EPS 画像の扱い

`Dvipsd mf x` の場合は基本的に PDF, JPEG, PNG, BMP, 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事になります。L^AT_EX の原稿中で `\includegraphics` 命令を用いて EPS 画像を張り込んでいる場合は、`Dvipsd mf x` が DVI から PDF への変換の段階で Ghostscript プログラムを毎回実行して EPS を EPDF に変換しています。そのため、`Dvipsd mf x` をデバイスドライバとして使用しているときには極力 EPS ではなく、EPDF 画像を張り込むようにします。外部プログラムが PDF での保存に対応していないようであれば、あらかじめ EPS を EPDF に変換すると処理速度の向上につながります。

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

る事がほとんどです。その時に `epstopdf` か `ps2pdf` などを 사용합니다^{*8}。 `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 がうまく反映されません (2006 年 4 月現在)。 以下のようなシェルスクリプト `eps2pdfs` を作成します。

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

`eps2pdfs` を 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` があつたとすればこれは `<file>.pdf` と `<file>.bb` が作成されます。

▼ 6.5.6 PDF 画像の切り抜きと BoundingBox

何かしらのプログラムで作成した PDF には余計な余白が含まれている事がしばしばあります。これを自動的に切り抜く方法の一つとして Heiko Oberdiek 氏による `pdfcrop` を使う事により、PDF 画像の余白の切り抜きを行う事ができます (要 `pdfTeX`, Perl, Ghostscript)。 使い方はコンソールから次のようにするだけです。

```
$ pdfcrop input.pdf
```

これにより `input-crop.pdf` が生成されます。

また、PDF の正確な BoundingBox を取得する一つの方法として `pdftinfo` を使う事が考えられます。

`pdftinfo` によって表示される情報は以下のような構成になっています。

```
「 Creator: TeX
  Producer: pdfTeX-1.10b
  CreationDate: Sat Apr 15 21:23:00 2006
```

^{*8} Vine Linux の場合は `ps2jpdf` という日本語フォントを埋め込まない PDF を作成できるプログラムもあります。 `apt-get update; apt-get install ps2jpdf` でインストールできます。


```

Tagged: no
Pages: 1
Encrypted: no
Page size: 416 x 40 pts
File size: 7995 bytes
Optimized: no
PDF version: 1.4

```

pdfcrop によって切り抜きを行った画像であれば、‘Page size: 416 x 40 pts’ を適切に加工すれば BoundingBox として使えるようになるでしょう。

以下のようなスクリプト makeebb を用意します*9。

```

#!/bin/sh
# 引数として与えられたディレクトリを作業対象とする
cd $1;
# PNG 画像の BoudingBox の生成を ebb により行う
for f in `ls *.png`; do
    if [ -f `basename $f .png`.bb ] ; then
        echo "already `basename $f .png`.bb exists."
    else
        ebb -v $f;
    fi
done
# JPEG 画像の BoundingBox の生成を ebb により行う
for f in `ls *.jpg`; do
    if [ -f `basename $f .jpg`.bb ] ; then
        echo "already `basename $f .jpg`.bb exists."
    else
        ebb -v $f;
    fi
done
# PDF 画像の BoundingBox の生成を pdftinfo によって 行う
for f in `ls *.pdf`; do
    if [ -f `basename $f .pdf`.bb ] ; then
        echo "already `basename $f .pdf`.bb exists."
    else
        echo "creating `basename $f.pdf`.bb..."
        pdftinfo $f | grep -e 'Page size:' | \
        sed -e 's/x//; s/Page size:\/\%\/\%BoundingBox: 0 0 /; s/pts//;' \
        > `basename $f .pdf`.bb
    fi
done

```

*9 <http://tex.dante.jp/joul/makeebb>

これを例えば、適当にアクセス権を与えて `makebb img` とすれば、`img` ディレクトリに存在する PNG, JPEG, PDF 画像の BoundingBox を作成します。

ストリームエディッタの `sed` がない場合は適当に Perl 等で実行してください。

▼ 6.5.7 dvips と Dvipdfmx の併用

`dvipsk` と `Dvipdfmx` の両方を併用している (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/gnu-head}
```

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、`dvipdfmx` 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[dvipdfmx]{graphicx} % Dvipdfmx の場合
```

▼ 6.5.8 レポート・論文における図の張り込み

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、次のように使う事になります。

```
\begin{figure}[htbp]
  \begin{center}
    \includegraphics[width=10cm]{images/file.eps}
    \caption{図見出し}\label{fig:samplefig}
  \end{center}
\end{figure}
```

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\linewidth]{%
  \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簡単に使えます。

```
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が配置されなくても腹を立てないでください。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。ですから本来は図表がどのような場所に旅立っても困らないはずです。

▼ 6.5.9 汎用的な画像の作成と活用

LaTeX と Dvips/dvips を用いる事で、JPEG、PNG、BMP、EPS、PDF 等の画像を張り込む事が可能でした。しかし、外部プログラムによってはそれらの形式の画像ファイルの書き出し（変換）に対応していない場合があります。この場合はある特定のプログラムから、仮想プリンタに対して画像の内容を送信し、EPS か PDF で保存するのが手短にできる方法となります。

Windows であれば PrimoPDF 等のフリーの変換プログラムがあります。Mac OS X であれば OS そのものが PDF での書き出しに対応しています。

現在お使いの環境に Adobe Acrobat がある場合は、Acrobat を活用していただいで構いません。

▼ 6.5.10 プログラム特有の処理

特定の外部プログラム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。6.5.9 節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い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*10。

Illustrator 可能であれば文字はアウトライン化します。Adobe PDF の互換性では

*10 プログラムのバージョンによっては幾分操作方法が異なると思います。

[Acrobat 4 (PDF 1.3)]を指定するようにすると、問題が発生しづらいつと思われる。ツールバーの[別名で保存]でファイル形式を‘Adobe PDF’として保存します。PDF 形式での保存オプションで「サムネールを埋め込み」の**チェックを外して**、「圧縮」はしないようにしてください。Illustrator の場合は用紙サイズが切り抜かれませんので何らかの方法 (Adobe Acrobat や `\includegraphics` 命令の `trim` オプション) で切り抜きを行う必要があります。

Photoshop [ファイル], [複製を保存]を選び「保存形式」を‘Photoshop PDF’にして保存します。ビットマップ画像は圧縮しないほうが印刷品質が良いようです。

Gnuplot フリーのプロットソフトで PostScript, PSTricks, Tgif, Illustrator, eepic, METAFONT, METAPOST 等, 多くの形式で画像の書き出しをサポートしています。Octave も MATLAB 類似で GPL の数値演算ソフトで Gnuplot をもとに開発されていますので手順は Gnuplot の場合とほとんど同じです。eepic パッケージで対処するには、例えば Gnuplot 側で次のようにします。

```
set output 'plotfile1.tex'\\
set term eepic rotated dashed\\
plot x
```

すると、カレントディレクトリに `plotfile1.tex` が作成されますから、eepic パッケージ等を用いて、L^AT_EX の原稿側で次のように記述します。

```
\documentclass[dvipdfmx]{jsarticle}
\usepackage{graphicx,color,epic,eepic,amssymb}
\begin{document}
\input{plotfile1}
\end{document}
```

この場合は `graphicx`, `epic`, `eepic`, `amssymb` パッケージを必要としており、`\input` 命令でプロットされたグラフ `plotfile1.tex` を読み込むようになっています。

R GPL の統計解析ソフトで PostScript, PDF, PicT_EX, Xfig, PNG, JPEG 等の書き出しをサポートしています。

```
pdf()\\
plot(rnorm(10))\\
dev.off()
```

上記のように R から操作すればカレントディレクトリに PDF 形式のグラフ `Rplots.pdf` が作成されます。

Tgif William Chia-Wei Cheng 氏による QPL の描画ソフト。EPS や PDF 形式に対応しています。PDF に関しては Ghostscript 等の外部プログラムを必要とします。

Mac OS X Mac OS X の場合は環境自体が PDF に関連した機能を持っているため、PDF 形式で書き出す事により \LaTeX に画像を取り込む事ができます。Keynotes, Pages, Grapher, OmniGraffle 等、いずれの場合もメニューバーの [ファイル] の [書き出し] で [PDF] を選択する事で PDF として保存できます。プレビューで [選択ツール] によって切り抜きたい領域を選択し、それを [コピー] した後 [クリップボードから新規作成] とすれば PDF 画像の切り抜きもできます。

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事があるので \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次のような記述があります。

```
%%BoundingBox: 91.5625 3.1875 321.938 190
```

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。

一般的には PDF にさえ変換していれば Adobe Acrobat による編集が可能となり、さらに Dvipdfmx を用いれば簡単に画像を張り込む事ができます。一度 PDF に画像を変換すると、その PDF ファイルの編集は Adobe Acrobat のような PDF 編集プログラムが必要となります。そのため、画像の調整に関しては元の外部プログラム側で行うようにしてみてください。

6.6 図の張り込みの際の工夫

▼ 6.6.1 図を二つ横に並べる

2 段組の場合はそのような事はありませんが、1 段組の場合是一个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
    {\small (a) 初期値 $c=0.6$ }
  \end{minipage} \hfill
  \begin{minipage}{.47\textwidth}
    \centering%ここに図 (b)を入れる
    {\small (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。二つ以上横に並べるとき等には Steven Douglas Cochran 氏による `subfigure` パッケージを使うとより簡単に記述できる事になります。



(a) 初期値 $c = 0.6$



(b) 初期値 $c = 1.0$

図 6.3 1 段組で横に図を二つ並べる

▶ **問題 6.5** ある環境などにおけるその時々文章幅を保持している `\linewidth` という長さがあります。この長さを使うとその環境において文章幅いっぱいの図を張り込むという事もできるようになります。

次の入力を実際に自分でタイプセットし、その結果を吟味してください。

```

\begin{quote}
  linewidth $=$ \the\linewidth\par
\begin{quote}
  linewidth = 135.81981pt
  linewidth $=$ \the\linewidth
  linewidth = 117.32607pt
\end{quote}
\end{quote}

```

これにより行の半分程度の長さで図を張り込むならば、次のように設定できると考えられるでしょう。

```
\includegraphics[width=.47\linewidth]{images/gnu-head}\
```

▶ **問題 6.6** 図表を張り込む時に `\includegraphics` 命令に毎回ディレクトリ名を記述するのが面倒な場合は `\graphicspath` 命令が使えます。

```
\graphicspath{<ディレクトリのリスト>}
```

仮に `images` と `pictures` ディレクトリに画像が保存されているとすれば、`\graphicspath` は次のようにできるか、実際にタイプセットし、その結果を吟味してください。

```
\graphicspath{{images/}{geolay/}}
```

▼ 6.6.2 画像に文字を追加する——`labelfig`

再編集が難しい画像ファイル、例えば EPS ファイルの上に文字などのラベルを追加したい場合があります。これには Raymond Séroul 氏と Laurent Siebenmann 氏による `labelfig` パッケージが使えるでしょう。

```

\SetLabels
<画像の上に表示したいラベル>
\endSetLabels
\ShowGrid (必要に応じて)
\strut\AffixLabels<配置する画像>

```

`\SetLabels` から `\endSetLabels` の中で画像の上に表示したいラベルを設定します。ラベルを追加するときに必要に応じて `\ShowGrid` コマンドで座標を表示します。

`\AffixLabels` の引数に配置すべき画像を指定します。ラベルは次の書式に従って追加します。

```
⟨位置指定⟩(⟨0-1⟩*⟨0-1⟩) ⟨ラベル⟩ \\\
```

座標指定は $(0.5*0.3)$ のように 0 から 1 の範囲で指定します。⟨位置指定⟩には垂直方向の揃えでは `\T`, `\E`, `\B`, 水平方向では `\L`, `\R` と無指定（無指定で中央になる）の両方を組み合わせて使う事ができます。

`\ShowGrid` によってグリッドを表示するのは原稿執筆段階だけで、印刷時には表示しないとすれば `draft` オプションを活用します。ただし、`graphicx` パッケージによって読み込んでいる画像に関しては `draft` オプションが有効になっているときでも `final` オプションを付けたときのように配置してもらいたいので、例えば次のようにします。

```
% グリッドを表示させるのは draft の時だけにすれば良いことになる
%\documentclass[draft,a4j,11pt,papersize]{jsarticle}
% 印刷時には draft オプションを除けば良いことになる。
\documentclass[a4j,11pt,papersize]{jsarticle}
% graphicx パッケージには final を渡して、いつでも図が表示される
% ようにすると, labelfig の調整が容易になる。
\usepackage[final]{graphicx}
\usepackage{labelfig}
```

例えば次のような入力があれば 図 6.4 のような出力になります。 `\GridLineWidth` コマンドで罫線の太さを指定できます。

```
\begin{figure}[htbp]
\begin{center}
\GridLineWidth{.2pt}
\SetLabels
\T\L(.8*.45) 鼻\\
\T\L(.2*.9) 左の角\\
\T\L(.7*.9) 右の角\\
\T\L(.75*.3) □\\
\T\L(.65*.1) 髭\\
\T\L(.3*.6) 左目\\
\T\L(.7*.6) 右目\\
\endSetLabels
\ifdraft
\ShowGrid
\fi
\strut\AffixLabels{\includegraphics{images/gnu-head}}%
\caption{\Y{labelfig} の使い方\label{fig:you}}%
\end{center}
\end{figure}
```

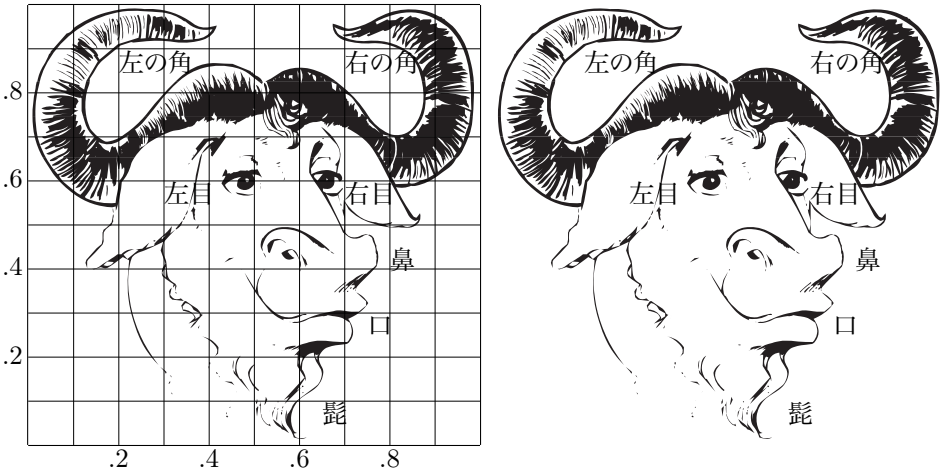



図 6.4 labelfig の使い方

6.7 描画の方法

\LaTeX 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 6.4 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 \LaTeX 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 (デバイスドライバ等) へ描画をゆだねる方法です。一般に \LaTeX における描画の能力は \TeX 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 \LaTeX に備わっている `picture` 環境による描画を行うのが手軽です。

▼ 6.7.1 ベタ書きによる図の作成

もっとも簡単な描画の方法として \LaTeX でベタ書きを行う、`verbatim` 環境を使う事が考えられます。 `verbatim` 環境内では文字が等幅に近い字詰めで組まれるので、原稿で入力している表示と DVI ファイルへの出力が同じようになります^{*11}。

^{*11} 内田昭宏氏の作成した `plain2` というツールを使うと全角記号を組み合わせる事によって \LaTeX 用の図表を作成することもできます。

▷ **例題 6.7** 以下の入出力例を参考にべた書きによる作図をしてみてください。

```
\begin{verbatim}
      a
     / \
    /   \
   b     c
  / \   / \
 d e f g
\end{verbatim}
```

```
      a
     / \
    /   \
   b     c
  / \   / \
 d e f g
```

▼ 6.7.2 曲線の描画

ベクトル画像などではベジェ曲線とかスプライン曲線という近似曲線が使われていると多くの参考書で記述されています。ベクトル画像を知るうえでベジェ曲線の原理を知っておくと、曲線を描くときに頭の中で曲線をイメージしやすいと思いますので紹介しておきます。滑らかな曲線を描くためには多くの点座標が必要になると思う人もいるでしょうが、ある程度滑らかな曲線を描くためには3点あれば十分です。曲線を描くための点（制御点）が少なければ少ないほど情報量は少なくなるので、少ない制御点で滑らかな曲線を描く方法が過去に模索されました。その中でもベジェ曲線は高々二つの基準点と一つの制御点（2 + 1点）があれば現在私たちがIllustratorなどでよく見かける曲線になります。この原理がIllustratorのペンツールに活かされていますので、お持ちの方は確認していただくと良いでしょう。ただIllustratorの場合はユーザの見えない箇所で様々な工夫がなされています。

曲線を描くためにいま n 個の制御点がありその i 番目の座標を $P_i = (x_i, y_i)$ として式 6.1 と式 6.2 で表す曲線をベジェ曲線と呼びます。

$$P(u) = \sum_{i=0}^{n-1} P_i B_{i,n}(u) \quad (6.1)$$

$$B(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad (6.2)$$

これがベジェ曲線の一般形ですが、例としてType1フォントでも使われている？ 2次ベジェ曲線を示します。平面座標に $P_0 = (-1, 1)$, $P_1 = (0, 0)$, $P_2 = (1, 1)$ があるとすれば式 6.1 と式 6.2 より、

$$\begin{aligned} P(u) &= P_0 B_{0,2} + P_1 B_{1,2} + P_2 B_{2,2} \\ &= P_0 (1-u)^2 + P_1 2u(1-u) + P_2 u^2 \end{aligned} \quad (6.3)$$

となり式 6.3 に対して無数の u を与えれば滑らかな曲線を描けます。これは 3 次元でも同様に計算できるので便利な式です。例の基準点、制御点とベジェ曲線は図 6.5 の通りになります。このような原理を知っておくと後ほど紹介する L^AT_EX の `picture` 環境

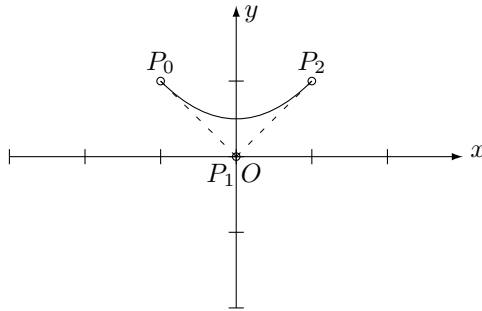


図 6.5 制御点と式から得られるベジェ曲線

で使用できる命令の理解に役立つ事でしょう。ただし L^AT_EX での多くのベジェ曲線を描くコマンドはもっと計算の少ないアルゴリズムを使っている場合がありますし、デバイスドライバに描画を任せている事もあります。

▼ 6.7.3 `picture` 環境による描画

L^AT_EX の力を使った描画を行うには特別な環境、描画専用の `picture` 環境で作業を使います。`picture` 環境では基準となる長さを決めてその相対的な距離によって描画を行います。このとき基準となる長さ `\unitlength` を決めます。

```
\begin{picture}(x,y)(x_0,y_0)
<描画内容>
\end{picture}
```

`picture` 環境の中に描画したい内容を記述します。`picture` 環境に渡す `(x,y)` は必須引数ですが `(x_0,y_0)` は任意引数です。`(x,y)` には座標における `picture` 環境の大きさを横方向は x で縦方向は y で指定します。これには単位などを付けずに数値で指定します。`(x_0,y_0)` には原点の位置を指定します。

何らかの要素を配置するには `\put` か `\multiput` を使います。 x と y は単位 `\unitlength` に従属します。

```
\put(x,y){要素}
\multiput(x,y)(Δx,Δy){回数}{要素}
```

`\put` 命令は座標 (x, y) に〈要素〉を置くだけの命令です。 `\multiput` は座標 (x, y) を基点とし、二つ目の座標 $(\Delta x, \Delta y)$ をベクトルとして $(\Delta x, \Delta y)$ の変化量に応じて要素を回数分だけ繰り返して配置します。この他に 2 次ベジェ曲線を描く `\q bezier` 命令があります。

```
\q bezier(x1,y1)(x2,y2)(x3,y3)
```

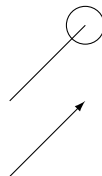
‘ (x_1, y_1) ’ を始点、‘ (x_2, y_2) ’ を基準点、‘ (x_3, y_3) ’ を終点として 2 次ベジェ曲線を描きます。

〈要素〉には次のようなコマンドが標準として使えます。

```
\line(x,y){長さ}
\vector(x,y){長さ}
\circle*{直径}
\oval(幅,高さ)[(位置指定)]
```

`\line` は ‘ (x, y) ’ をベクトルとして〈長さ〉分の線分を描きます。 `\vector` は `\line` の終端に矢印をつけたものです。 `\circle*` は直径を指定して円を描きます。アスタリスク ‘*’ を付けないと円の内側が塗りつぶされません。 `\oval` は幅と高さを指定して楕円を描きます。

```
\setlength{\unitlength}{1mm}
\begin{picture}(40,30)
\put(10,10){\line(1,1){10}}
\put(10,0){\vector(1,1){10}}
\put(20,20){\circle{5}}
\end{picture}
```



楕円を描く `\oval` 命令の任意引数の〈位置指定〉には楕円のどの部分を出力するかを指定します。それぞれ上部 ‘t’、下部 ‘b’、左 ‘l’、右 ‘r’ となり、複合的に使用できます。

```

\setlength{\unitlength}{1mm}
\begin{picture}(50,30)
\put(8,12){\oval(10,15)[t1]}
\put(8,8){\oval(10,15)[b1]}
\put(10,10){\oval(10,15)}
\put(12,12){\oval(10,15)[tr]}
\put(12,8){\oval(10,15)[br]}
\end{picture}

```

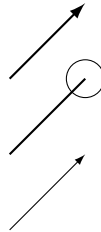


picture 環境中での線の太さは `\thinlines` と `\thicklines` の二つで調整します。 `\thinlines` のほうが細く `\thicklines` のほうが太くなります。 picture 環境中の全ての線分に有効になります。

```

\setlength{\unitlength}{1mm}
\begin{picture}(50,30)
\thicklines
\put(10,10){\line(1,1){10}}
\put(10,20){\vector(1,1){10}}
\thinlines
\put(10,0){\vector(1,1){10}}
\put(20,20){\circle{5}}
\end{picture}

```



▷ **例題 6.8** 図 6.5 を picture 環境で描画するにはどうすれば良いかを考えてください。例を示すと以下のような記述になります。

```

\setlength{\unitlength}{1cm}
\begin{picture}(6,4)(0,0)
\put(0,2){\vector(1,0){6}}
\put(6.2,2){\makebox(0,0)[b]{\$x\$}}
\put(3,0){\vector(0,1){4}}
\put(3.2,4){\makebox(0,0)[t]{\$y\$}}
\multiput(0,1.9)(1,0){6}{\line(0,1){.2}}%後述
\multiput(2.9,0)(0,1){4}{\line(1,0){.2}}%後述
\put(2,3){\circle*{.1}}
\put(2,3.1){\makebox(0,0)[b]{\$P_0\$}}
\dashline{.1}(2,3)(3,2)%後述
\put(3,2){\circle*{.1}}
\put(2.8,1.9){\makebox(0,0)[t]{\$P_1\$}}
\put(3.2,1.9){\makebox(0,0)[t]{\$0\$}}
\put(4,3){\circle*{.1}}
\put(4,3.1){\makebox(0,0)[b]{\$P_2\$}}
\dashline{.1}(4,3)(3,2)%後述

```

```
\qbezier(2,3)(3,2)(4,3)
\end{picture}
```

▼ 6.7.4 picture 環境の拡張その 1——epic

LaTeX での標準の `picture` 環境のコマンドもデバイスに依存しないので汎用性があるのですが、それではあまりにも表現力に乏しいのが現状です。そこでこの `picture` 環境の拡張が行われてきました。`picture` 環境に限らず LaTeX での描画は 1980 年代後半からさまざまな方法が模索され、拡張され続けました。その中でも Sunil Podar 氏による `epic` は `picture` 環境の拡張版としては定評があります。`epic` では LaTeX の `picture` 環境で使用できるコマンドのほかに以下の命令が拡張されています。

```
\multiputlist \matrixput \grid \picsquare \dottedline
\dashline \drawline \jput \putfile
```

このほかに `dottedjoin`, `dashjoin`, `drawjoin` の三つの環境が定義されています。座標の変化量を $(\Delta x, \Delta y)$ として複数の項目を配置する `\multiputlist` 命令があります。

```
\multiputlist(x,y)(\Delta x,\Delta y)[\langle tbrl \rangle]{\langle 複数の項目 \rangle}
```

座標上に行列のように要素を繰り返して配置する `\matrixput` 命令もあります。

```
\matrixput(x,y)(\Delta x_1,\Delta y_1){\langle n_1 \rangle}(\Delta x_2,\Delta y_2){\langle n_2 \rangle}{\langle 要素 \rangle}
```

```
\setlength{\unitlength}{1pt}
\begin{picture}(150,110)(0,0)
\multiputlist(0,0)(15,10)%
{0,1,2,3,4,5,6,7,8,9,10}
\matrixput(0,0)(20,0){7}(0,20)%
{5}{\mbox{未来}}
\end{picture}
```

10
9
8
7
6
5
4
3
2
1
0

座標系を表現するのに格子を描くには `\grid` 命令が使えます。

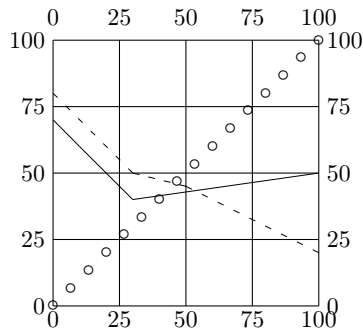
```
\grid(幅,高さ)(\Delta幅,\Delta高さ)[\langle x 座標の初期値,y 座標の初期値 \rangle]
```

他にも点線や破線などを描くコマンドがあります。

```
\dottedline[⟨点の種類⟩][⟨間隔⟩](x1,y1)(x2,y2)⋯(xn,yn)
\dashline{⟨破線の長さ⟩}[⟨間隔⟩](x1,y1)(x2,y2)⋯(xn,yn)
\drawline(x1,y1)(x2,y2)⋯(xn,yn)
```

`\dottedline` は点線を、`\dashline` は破線を、`\drawline` は折れ線を描くために使います。点線や破線は折れても構いません。

```
\setlength{\unitlength}{1pt}
\begin{picture}(150,120)(0,0)
\put(0,0){%
  \grid(100,100)(25,25)[0,0]}
\dottedline[ $\circ$ ]{10}(0,0)
  (100,100)
\dashline{3}(0,80)(30,50)(50,45)
  (100,20)
\drawline(0,70)(10,60)(30,40)
  (100,50)
\end{picture}
```



▼ 6.7.5 picture 環境の拡張その 2——epic

Sunil Podar 氏による epic を改良・拡張したものとして、Conrad Kwok 氏の作成した epic があります。これは epic の改良・拡張版でありますので使用するときには次のように epic も先に読み込んでおきます。

```
\usepackage{epic,epic}
```

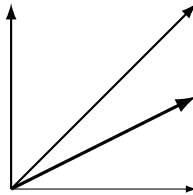
LaTeX の picture 環境で使用できる `\line`、`\circle*`、`\oval` の拡張が行われています。epic のコマンドも全て再定義されています。epic はこれらの命令を TPIC スペシャルに置き換えていますので描画力は高いのですが、デバイスドライバが TPIC スペシャルに対応している必要があります。dviout、dvips、Dvipdfmx などは対応しているようです。デバイスドライバによって TPIC スペシャルの解釈が若干異なるようですので、出力を確認してデバイスを選択してください。

線の太さに関するコマンドが新たに定義されています。

```
\allinethickness{⟨太さ⟩}
\Thicklines
```

`\allinethickness` はこの命令を使った後の `picture` 環境中にある全ての線の太さを変更します。 `\Thicklines` は `\thicklines` よりもこのコマンドを宣言した後の線の太さを太くします。

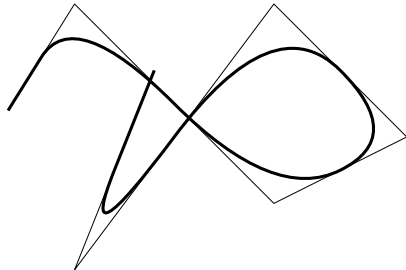
```
\begin{picture}(100,100)(0,0)
\newcommand\VP[2]{%
  \put(0,0){\vector(#1,#2){70}}}
\thinlines \VP{1}{0}
\thicklines \VP{1}{1} \VP{0}{1}
\Thicklines \VP{2}{1}
\end{picture}
```



`epic` では `\drawline` よりも `\path` を使い、 `\qbezier` よりも `\spline` を使うと良いでしょう。 `\spline` は始点と終点以外は制御点として Chaikin 曲線を描きます。

```
\path(x1,y1)(x2,y2)⋯(xn,yn)
\spline(x1,y1)(x2,y2)⋯(xn,yn)
```

```
\setlength{\unitlength}{1pt}
\begin{picture}(150,100)(0,0)
\path(0,60)(25,100)(100,25)
(150,50)(100,100)(25,0)(55,75)
\Thicklines
\spline(0,60)(25,100)(100,25)
(150,50)(100,100)(25,0)(55,75)
\end{picture}
```



卵形の楕円を描くのに `\ellipse` を、弧を描くには `\arc` を使います。アスタリスクを付けると領域を塗りつぶします。

```
\ellipse*{⟨幅⟩}{⟨高さ⟩}
\arc{⟨長さ⟩}{⟨始点角度⟩}{⟨終点角度⟩}
```

⟨始点角度⟩の値は $[0, \pi/2]$ の範囲に、⟨終点角度⟩の値は $[\text{始点角度}, \text{始点角度} + 2\pi]$ の範囲にします。領域の塗りつぶしには `\filltype` を使います。アスタリスクを付けた場合の `\circle*` と `\ellipse*` の領域の塗りつぶす種類を `'black'`、`'white'`、`'shade'` の三つから選択します。

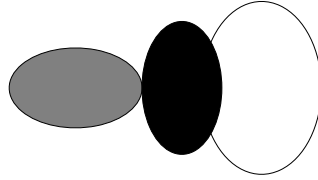
```
\filltype{⟨種類⟩}
```



```

\begin{picture}(150,100)(0,0)
\filltype{shade}
\put(10,50){\ellipse*{50}{30}}
\filltype{black}
\put(50,50){\ellipse*{30}{50}}
\put(80,50){\ellipse{45}{65}}
\end{picture}

```



▼ 6.7.6 picture 環境の拡張その 3——pict2e

Hubert Gäblein 氏と Rolf Niepraschk 氏による `pict2e` は `picture` 環境の拡張として 2003 年頃に公表されたものです。デバイスドライバの力を借りて `picture` 環境で使用できるコマンドを再定義していますし、新しいコマンドも定義されています。今の所

```
dvips xdvi pdftex vtex dvipdfm
```

などのデバイスドライバをサポートしています。`picture` 環境におけるほとんどのコマンドを拡張してあります。`\circle` で描ける円の大きさにも制限はありません。ベジェ曲線に関しては `\cbezier` 命令が追加されました。

```

\qbezier(x1,y1)(x2,y2)(x3,x3)
\cbezier(x1,y1)(x2,y2)(x3,x3)(x4,x4)

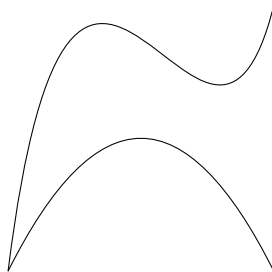
```

`\qbezier` は 2 次ベジェ曲線用、`\cbezier` は 3 次ベジェ曲線用の命令です。

```

\setlength{\unitlength}{1pt}
\begin{picture}(100,100)(0,0)
\qbezier(0,0)(50,100)(100,0)
\cbezier(0,0)(25,200)
(75,0)(100,100)
\end{picture}

```



6.8 他のプログラムによる描画

広く使われている描画ツールを紹介します。以下の描画ツールで作成した図などはそれぞれ指定された方法でデバイスドライバが適切に解釈できる状態になればなりま

せん。

▼ 6.8.1 TPIC による描画

Unix 系 OS の描画ツールとして Brian Kernighan 氏が開発した PIC があります。これを Tim Morgan 氏が $\text{T}_{\text{E}}\text{X}$ に移植した TPIC を使うと簡単な図形ならばすぐに描くことができます。Unix 系 OS を持っているならば PIC というプログラムが導入されているでしょう。ただ PIC は日本語化されていないかもしれないので注意してください。TPIC はほとんどの機能を $\text{T}_{\text{E}}\text{X}$ ではなくデバイスドライバに任せています。これらの命令は `\special` 命令の中に記述されています。TPIC の出力する特有な命令を TPIC スペシャルと言います。TPIC スペシャルに対応しているドライバは `dvips` や `dviout`, `Dvipdfmx` などです。多くのドライバが対応していますが、デバイスドライバによる解釈の違いなどもありますので若干注意が必要でしょう。

▷ **例題 6.9** 例えば図 6.6 のような図を作成したいとしましょう。PIC の入力ファイル

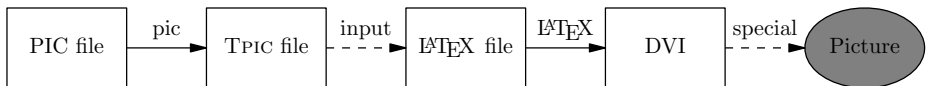


図 6.6 TPIC の使用例

`tpicsmpl.pic` に以下のような記述をします。

```
.PS
box "PIC file"
arrow "pic" above
box "\textsc{Tp}ic file"
arrow "input" above dashed
box "{\LaTeX} file"
arrow "{\LaTeX}" above
box "DVI"
arrow "special" above dashed
ellipse "Picture" fill
.PE
```

この `tpicsmpl.pic` を `pic` で $\text{T}_{\text{E}}\text{X}$ 用に出力するために `-t` オプションを付けて

```
$ pic -t tpicsmpl.pic > tpicsmpl.tex
```

とする事で `tpicsmpl.tex` 中の `\graph` に図形が格納されます。これを `file.tex` で

使用するためには次のように `\input` 命令でファイルを読み込み、`\box` 命令で実際に図を使用します。

```
\input{tpicsmpl}
\begin{center}
  {\box\graph}
\end{center}
```

するとその場所に中央揃えでグラフを挿入できます。適宜 `figure` 環境に入れるなどします。

▶ **問題 6.10** 例題 6.9 で作成したファイル `tpicsmpl.tex` の中を直接閲覧してみてください。するとどのような記述があるのかを確認できると思います。

基本的には数多くの `\special` 命令を使って、`pn`, `pa`, `fp`, `sh` をデバイスドライバに渡しているという事になっています。そこで、例えば `\special{pa 4500 0}` のような記述をいくつかコメントアウトしてみて、その命令がどのような役割を持っているのか、考えてみてください。

▶ **問題 6.11** Unix 系 OS であればコンソールから `man pic` として、どのような表現能力があるのかを確認してみてください。これにより、`for`, `if`, `sin`, `cos` 等のコマンドが使用できる事が分かります。

▼ 6.8.2 メタな描画プログラム

Donald Knuth 氏がフォントデザイン用に開発した METAFONT があります。これに対して John Hobby 氏が描画に関するアルゴリズムを追加したり、出力形式を EPS にした METAPOST という描画プログラムを開発しました。METAFONT は $\text{T}_{\text{E}}\text{X}$ のフォント形式ファイル `{file}.gf` を出力するのに対して METAPOST は EPS 形式ファイル `{file}.n` を出力するので今では METAPOST が広く使われています。METAPOST に関する日本語情報は少ないのが現状です。しかし METAFONT から変更・追加された箇所があったとしても、Donald Knuth 氏の *METAFONTbook* [54] が参考になると思います。

METAFONT をちょっと触ってみましょう。コンソールから

```
$ mf
```

とするとアスタリスク `*` が二つ表示され、ファイルの入力を促しています。

```
「
This is METAFONT, Version 2.7182 (Web2C 7.3.9)
**
」
```

ここでは実験的に、“\relax”と入力して改行します。するとアスタリスクが一つになるはずですが、

```
** \relax
```

これで準備は万全です。とりあえず点を表示してみましょう。

```
* drawdot (0,0); showit;
```

今度は直線を描くために

```
* draw (0,0)..(100,0); showit;
```

としてみましょう。曲線などは

```
* draw (0,100)..(100,100)..(100,0); showit;
```

とすると罫囲気がつかめるでしょう。終了するときには

```
* end.
```

と入力します。

今度は METAPOST を少し体験してみましょう。‘mp’ か ‘mpost’ をコンソールから実行すれば良いはずですが、今回は METAPOST のファイル hoge.mp を用意します。

```
beginfig(1)
u=100;
draw (u,u)--(2u,u)--(2u,2u)..cycle;
draw (u,u)..(2u,u)..(2u,2u)..(u,2u)..cycle;
endfig;
end.
```

これを見ても何がなんだかわかりませんが、とりあえず保存しておきます。コンソールから

```
$ mpost hoge.mp
```

とします。そうすると EPS 形式の hoge.1 が作成されますのでご覧ください。Ghostscript などで見ると図 6.7 のような円とその円に内接する直角二等辺三角形が表示されます。環境によっては日本語化された METAPOST を使うためには mpost ではなく

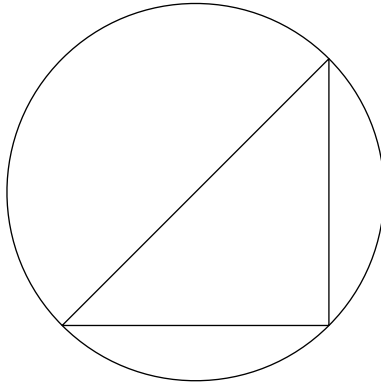


図 6.7 METAPOST の出力例

jmpost を使う事になるかもしれません。角藤亮氏の p_TE_X を使っている Windows の方は jmpost のはずです。

▼ 6.8.3 PSTricks

Timothy Zandt 氏らによる PSTricks は `\special` 命令中に PostScript 命令を記述する事によって PostScript の描画機能を T_EX/L^AT_EX で使えるようにするパッケージです。PostScript 命令を多用する事からデバイスドライバとして dvips などを想定していたり PostScript 対応プリンタでの使用が推奨されます。PSTricks の詳しい使い方は日本語訳で 70 ページ分の『L^AT_EX グラフィックスコンパニオン』[35] の第 4 章を参照してください。基本的なマクロを読み込むためには PSTricks パッケージを読み込みます。特定のパッケージを個別に読み込む事もできます。色を使うためには graphics パッケージに含まれる color パッケージではなく pst-col パッケージを読み込みます。全ての機能を使うときは pst-all を読み込みます。標準的に以下のようにするとうまく行くと思います。

```
\usepackage[dvips]{graphicx}
\usepackage[dvips,usenames]{pst-col}
\usepackage{pst-all}
```

具体的に三次元射影、画像の EPS 変換、グラデーション、木構造、回路図、プロットなどさまざまな事ができます。

PSTricks を Dvipdfmx で使うにはまず面倒な方法として PSTricks を使って描いた図形を一つ一つ EPS ファイルに変換する方法があります。

```
\documentclass{jsarticle}
\usepackage[dvips]{graphicx}
\usepackage[dvips,usenames]{pst-col}
\usepackage{pst-all}
\pagestyle{empty}
\begin{document}
\begin{pspicture}(100,100)
  描画内容
\end{pspicture}
\end{document}
```

このようなファイル `fig1.tex` を作成し、これをコンソールから次のようにします。

```
$ platex fig1
$ dvipsk -Pdl -E -o fig1.eps fig1
$ epstopdf fig1.eps
$ egrep "%BoundingBox:" fig1.eps > fig1.bb
```

上記の操作を図形の数だけこなし、図形を使用したい L^AT_EX の原稿に PDF ファイル `fig1.pdf` として `graphicx` の `\includegraphics` で張り込みます。

```
\usepackage[dvipdfmx]{graphicx}
\includegraphics[bb={0 0 100 100}]{fig1.pdf}
```

CV Radhakrishnan 氏と CV Rajagopal 氏らによる `pdftricks` を使うと少しは楽になります。これはもともと pdf_LTEX で PSTricks を使うためのものですが、pL^AT_EX にも流用できそうです。まずは `pdftricks` に付属の `pst2pdf` というシェルスクリプトを少し変更します。

```
for f in $FIGURES ; do
  latex $fig
  dvips -Ppdf -E -o $fig.eps $fig
  epstopdf $fig.eps
done
```

この部分で `'latex'` はもちろん `'platex'` にしますし、`'dvips'` もご自分のシステムに合うように変更してください。epstopdf で変換した PDF のバウンディングボックスは正常ですが、何かしらの問題があった時は次の 1 行も追加しておくとも良いでしょう。

```
egrep "%BoundingBox:" $FILE > $fig.bb
```

▼ 6.8.4 Xy-pic

ダイアグラムなどを描くには Kristoffer Rose 氏と Ross Moore 氏による Xy-pic パッケージを使うと良いでしょう。状態遷移図やオートマトン、回路図などを描く事ができ非常に洗練されたシステムになっています。詳細は『 \LaTeX グラフィックスコンパニオン』[35] の第 5 章を参照してください。

▼ 6.8.5 化学式・化学構造式

化学式や化学構造式を描くためには、藤田眞作氏による \Xy-pic パッケージを使うと良いでしょう。これは \LaTeX 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事ができます。 \Xy-pic について詳しく知りたい方は藤田眞作氏の書いた \Xy-pic [49] を参照してください。

▼ 6.8.6 グラフの描画

\LaTeX 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 (6.5.10 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*12。

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*13。

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。

*12 <http://takeno.iee.niit.ac.jp/~foo/gp-jman/>

*13 <http://www.ecl.sys.hiroshima-u.ac.jp/scilab/docjp.html>



第 7 章

文献一覧の作成

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事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 L^AT_EX での参考文献の取り扱い方を紹介します。

7.1 参考文献の明記

参考文献 (references) を明記する事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**事になります。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 Bib_TE_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 jBib_TE_X を使う事になると思わ

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。jBIBTEX 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、jBIBTEX 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。

7.2 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事が考えられます。そのときは `thebibliography` 環境を使います。文献を `\bibitem` 命令を用いて文書の末尾にまとめます。

```
\bibitem[<表示形式>]{<ラベル>} <項目>
```

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は `\cite` コマンドを使います。

```
<文献>~\cite[<注記>]{<ラベル>}
```

<注記>にはページ番号などを記述します。例を示すと以下のようになります。

```
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。複数の文献を参照する場合は文献~\cite{KK1981}
\cite{AY1991}とせず、文献~\cite{KK1981,AY1991}とするのが正しい。
ただし、注記がある場合は複数の参照を一つにまとめるという事はないという
規則も存在する~\cite[Chapter.~8]{KK1981}\cite[pp.80--89]{AY1991}.
```

```
\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]は一読したい名著である。複数の文献を参照する場合は文献 [1][2] とせずに、文献 [1,2] とするのが正しい。ただし、注記がある場合は複数の参照を一つにまとめるという事はしないという規則も存在する [Chapter. 8; 1][pp.80-89; 2].

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。

▼ 7.2.1 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、書籍の場合は例えば次のように統一するという方法があります。

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

読者には**誰のなんという文献**という事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。

7.3 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。L^AT_EX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jBibT_EX [88, 89] 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。

▼ 7.3.1 jBibT_EX の使い方

参考(引用)文献はL^AT_EX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よいのですが拡張子は.bibとなるようにしてください。

▼ 7.3.2 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` という事にしておきます。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 '@' からはじめます。
- '@' 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' である事が分かります。
- 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 7.2 節の場合と同様です。

▼ 7.3.3 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事はありません。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。例えば、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文献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> (\cite からラベル参照を作成)
$ jbibtex <file> ((文献一覧).bib を入力)
$ platex <file> ((文献一覧).bb1 からラベル情報を取得)
$ platex <file> (相互参照の問題が解決する)
```

すると文献一覧が出力されます。

`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` が異常終了した事を意味しますので、`JBiBTeX` のログファイル `<file>.blg` を参照してください。

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事になります。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。

さて、どうしてこのようになっているのかを少し考えてみましょう。まずは以下のようなファイル `hoge.tex` を作成してください。

```
\documentclass{jsarticle}
\begin{document}
この文書~\cite{TW2004a}は稚拙だ。
\bibliographystyle{jplain}
\bibliography{ref}
\end{document}
```

次に以下のような参考文献データベース `ref.bib` を作成してください。

```
@book{TW2004a,
author = {渡辺 徹},
yomi = {Toru Watanabe},
title = {未来を深く考える},
publisher = {日本北海出版},
year = {2004},
note = {実在しません}}
```

このようなファイルが出来上がったら端末から `hoge.tex` を 1 回だけタイプセットします。すると端末には次のような警告が表示されます。

```
「
LaTeX Warning: Citation 'TW2004a' on page 1 undefined on
input line 3.
No file hoge.bb1.
LaTeX Warning: There were undefined references.
」
```

一つ目の警告では参照する対象が見つからないと言われていました。次に `No file hoge.bb1` と言われて `hoge.bb1` というファイルが足りない事になっています。最後に正しく相互参照の処理をできなかったと報告されます。

ここで `hoge.aux` の中身をのぞいてみましょう。ファイルの中には以下の4行が書き出されているでしょう。

```
\relax
\citation{TW2004a}
\bibstyle{jplain}
\bibdata{ref}
```

実は `JBibTeX` はこの情報を使って参考文献の並び替えをします。例えば文書で引用された順に文献を並び替えるときにはその引用された順番が分からなければなりませんから、このように `<file>.aux` から何らかの情報を得る事になるのです。

次に `JBibTeX` を使って文献の一覧を作成します。端末などから

```
$ jbibtex hoge
```

とすると次のようなメッセージが表示されます。

```
┌
  This is JBibTeX, Version 0.99c-j0.33 (Web2C 7.5.2)
  The top-level auxiliary file: hoge.aux
  The style file: jplain.bst
  Database file #1: ref.bib
└
```

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 (`hoge.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。

このようにして並べ替えなどを終えた文献一覧はこの場合 `hoge.bbl` に書き出されています。実際 `hoge.bbl` の中身を見てみると次のような出力となっています。

```
\begin{thebibliography}{1}
\bibitem{TW2004a} 渡辺徹.
\newblock 未来を深く考える.
\newblock 日本北海出版, 2004.
\newblock 実在しません.
\end{thebibliography}
```

これは7.2節で手動で記述した場合と類似しています。この段階で `hoge.bbl` が作成されていない場合は何らかの記述ミスが考えられますので `JBibTeX` の処理結果を `hoge.blg` から読み取ってください。

うまく `hoge.bbl` が作成されているならば、その文献一覧を `hoge.tex` に取り込むために

```
$ platex hoge.tex
```


として再度タイプセットします。すると端末には次のような警告などが表示されます。

```

┌
LaTeX Warning: Citation 'TW2004a' on page 1 undefined on input line
  3.
(./hoge.bbl) [1] (./hoge.aux)
LaTeX Warning: There were undefined references.
LaTeX Warning: Label(s) may have changed.
Rerun to get cross-references right.)
└

```

ラベルに関して何か変更があったから再度タイプセットしなさいと言われていた
ので、

```
$ platex hoge.tex
```

として3度目のタイプセットをする事になります。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(`hoge`).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だと思います。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

```

$ platex hoge.tex
$ jbibtex hoge
$ platex hoge.tex
$ platex hoge.tex

```

とすれば良い事になります。



基本的に文献一覧(`file`).bb1 は文書(`file`).tex の巻末にまとめる事になりますから、本文から参照 (`\cite`) する場合は必ず後方参照 (ファイルの後ろに参照元がある状態) になってしまいます。そのため、`BIBTEX` を実行した後に2回ほど `LATEX` を実行しなければなりません。

▶ **問題 7.1** これによりファイルの後方を参照している時には、絶対2回以上のタイプセット (コンパイル) が必要になります。さらにCコンパイラが実は複数回に渡って同じファイル (`file`).c を読み込んでいるという事実が判明します。

```

\documentclass{jsarticle}
\begin{document}
\section{序論}\label{joron}
\section{本論}\label{honron}
\ref{joron}節では云々。
\end{document}

```

参照する要素が一行先にあるだけでも、 \TeX のような伝統的なプログラムは同じファイルを複数回読み込む必要があるわけです。

```
\documentclass{jsarticle}
\begin{document}
\section{序論}\label{joron}
\ref{honron}節では云々。% 後方参照を追加
\section{本論}\label{honron}
\ref{joron}節では云々。
\end{document}
```

▶ **問題 7.2** 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照していない (\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照しなくても文献一覧に出力したいときは \nocite コマンドを使います。

```
\nocite{<ラベル>}
```

適当な文献データベース $\langle file \rangle$.bib が存在するとして、以下のファイルをタイプセットし、この状況における \nocite 命令の役割について考えてみてください。

```
\documentclass{jsarticle}
\bibliographystyle{jplain}
\begin{document}
\nocite{*}
\bibliography{file}
\end{document}
```

▼ 7.3.4 文献の種類及び項目

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

$\text{\bibliographystyle}$ 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コンマで区切って読み込んでもできます。

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 7.1 から選んでください。

表 7.1 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

‘author’, ‘title’, ‘publisher’, ‘year’ 以外にも指定できる項目があります。文献リストの各文献に表 7.2 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 7.3 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べ替えに若干の影響が出ますが、それ程神経質になる必要はありません。

著者 ‘author’ が複数人数のときはコンマで区切るのではなく次のように ‘and’ を使用します。

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

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’ の人数が非常に多いときには名前を代表著者のみとします。

```
author={代表著者 and others}
```

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。

表 7.2 フィールド名

項目	内容
address	出版社の住所
annotate	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

▼ 7.3.5 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出力例により紹介します。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`, `abbrv`, `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事がありますので, それらを使用する事も可能です。

表 7.3 文献の種類における必須・任意項目

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

jplain 昇順に通し番号を付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「各著者のイニシャル 年号」で表示する. 'key' 項目を追加する事により表示するイニシャルなどを変更できる.

[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.

▼ 7.3.6 文献の追加例

▷ **例題 7.3** 文献データベースに書籍を追加する例です. 書籍 (book) 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事が必須となります. **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] L^AT_EX 2_ε 美文書作成入門』ならば, 次のようにします.

```
@book{bibunsyo,
  author = {奥村 晴彦},
  yomi = {Haruhiko Okumura},
  title = {[改訂第3版] \LaTeX\ 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 7.4** 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. 大澤英一氏らによる論文誌 *The RoboCup Synthetic Agent Challenge 97* を追加するには次のようにします.

```
@inproceedings{E01997,
  author = {Ei-ichi 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) という事で `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に来る執筆者) 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 7.5** 近年は WWW 上に存在する資料を参照する機会が多くなっているようです。このとき、参照資料のウェブページを記述する事があると思います。この場合は **URL** (howpublished), **閲覧日** (year, month), **題名** (title), **著者名** (author) を記述する事になります。奥村晴彦氏によって管理されている“`TeX Wiki`”というウェブページを参照するには次のようにします。

```

@misc{H02006,
  howpublished = {\url{http://oku.edu.mie-u.ac.jp/~okumura/texwiki
    /}},
  author = {奥村 晴彦},
  yomi = {Haruhiko Okumura},
  title = {{\TeX\ Wiki}},
  year = 2006,
  month = 2,
}

```

ここでは閲覧日を著者名の更新日としています。さらに `url` パッケージに含まれる `\url` 命令を使っていますので、詳細は 8.18 節を参照してください。“`TeX Wiki`”という文字ではバックスラッシュが含まれており、正しく処理できない場合がありますので、波括弧で全体をくくります。

例題 7.3–7.5 の二つの文献を `JBTeX` によって処理した結果、次のように並び替えられた文献一覧 (`file`).`bb1` が作成されます。

```

\begin{thebibliography}{9}
\bibitem{H02004} 奥村晴彦.
  \newblock [改訂第 3版] \LaTeXe\ 美文書作成入門.
  \newblock 技術評論社, 2004.
  \newblock 021.49/Ok.
\bibitem{H02006} 奥村晴彦.

```

```

\newblock {\TeX\ Wiki}.
\newblock \url{http://oku.edu.mie-u.ac.jp/~okumura/texwiki/},
  2 2006.
\bibitem{E01997} Ei-ichi 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`(file).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] $\text{\LaTeX} 2_{\epsilon}$ 美文書作成入門. 技術評論社, 2004. 021.49/Ok.
- [2] 奥村晴彦. \TeX Wiki. <http://oku.edu.mie-u.ac.jp/~okumura/texwiki/>, 2 2006.
- [3] Ei-ichi 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。

▼ 7.3.7 文献の複数参照——cite

`\cite{ラベル1, ラベル2, ..., ラベルn}` のように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。

▼ 7.3.8 参照の形式を変更する

ある分野では丸括弧を使い著者名と年号を使うスタイル「(渡辺 1999)」が推奨される(丸括弧は全角で)場合は

```
\makeatletter
\renewcommand{@cite}[2]{({#1\if@tempswa , #2\fi}) }
\renewcommand{@biblabel}[1]{(#1) }
\makeatother
```

この記述をプリアンブルに入れると良いでしょう。jBIBTEX を使っている場合は文献スタイルに依存します。親切な方がウェブで提供していると思いますので文献スタイルファイルを探してみてください。

もう少し細かい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 並べ替えを行いません。

などが用意されています。

設定できるコマンドとして表 7.4 の五つがあります。

表 7.4 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

まずは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います。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）という記述をしておけば良いのです。

```

\renewcommand\citimid{;}
\renewcommand\citeleft{()}
\renewcommand\citeright{}
\renewcommand\citepunct{,}

```

すると「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次のようにします。

```

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

```

すると「そうです (i-iii,v,vi; p. 130)」のようになります。

7.4 文献の管理

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。

Morten Alver 氏と Nizar Batada 氏が作成した JabRef があります*1。こちらは Java で動作するプログラムです。日本語がうまく通るかわかりませんが、ソースが公開されているのでどうにかなるでしょう。

シェアウェアで Ref for Windows があります*2。こちらは Windows 上で動作する文献管理プログラムです。Ref for Windows で文献のリストを作成し、それを `<file>.bib` に出力するという手法のプログラムです。

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

*2 <http://homepage3.nifty.com/refwin/>

第 8 章

L^AT_EX の応用

以下に示すコマンドなどはレポートや論文には必要不可欠という程の要素ではありませんが、いざという時に役に立つかもしれませんので、簡単に説明しておきます。

8.1 ページレイアウトの簡単な設定

▼ 8.1.1 版面のレイアウト

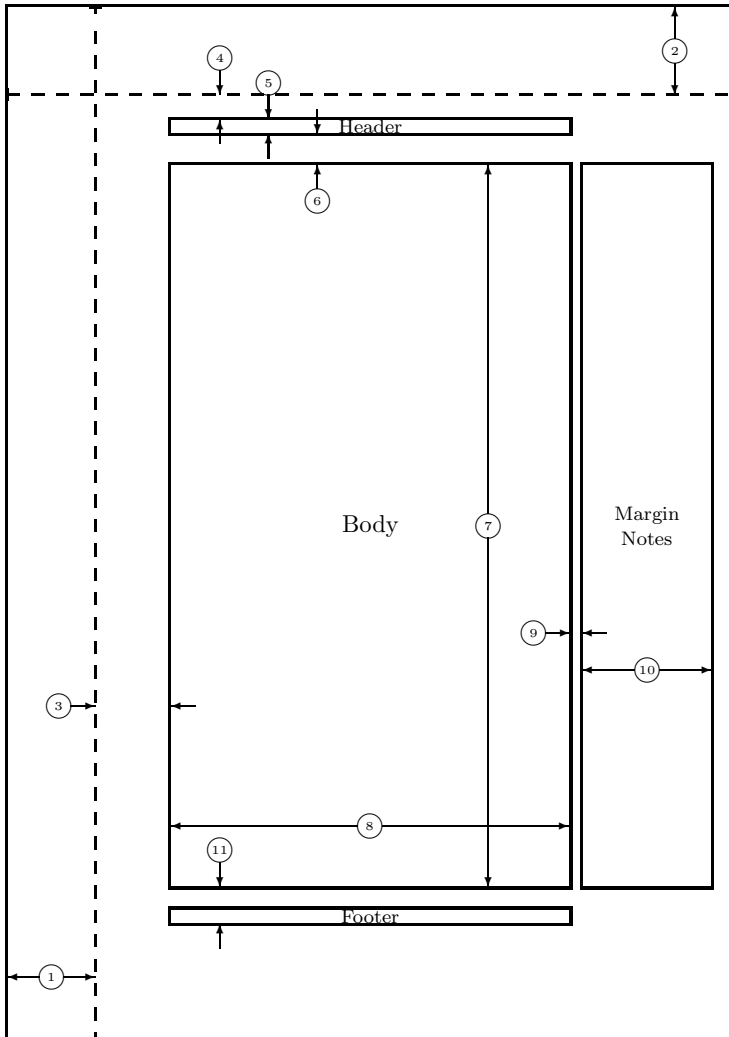
版面のレイアウトを行う場合にはそれぞれの長さに対して直接値を代入する方法があります。L^AT_EX で一般的に設定できる版面を調節する長さは図 8.1 の通りです。このような版面を視覚的に確認するには `layout` パッケージが使えます。このパッケージは使用されているクラスファイルから版面のレイアウトを出力します。使用方法は `document` 環境中で `\layout` 命令を使うだけです。

▶ **問題 8.1** `layout` パッケージを使って、特定のクラスファイルの標準的なページレイアウトがどのようになっているかを、次のようなファイルをタイプセットする事により確認してください。

```
\documentclass{jsarticle}
\usepackage{layout}
\begin{document}
\layout
\end{document}
```

`jsarticle` 以外にも `jsbook` や `jbook` 等でも確認してみてください。

まずはページ全体の余白に関する長さです。



- | | | | |
|----|------------------------------------|----|---|
| 1 | <code>one inch + \hoffset</code> | 2 | <code>one inch + \voffset</code> |
| 3 | <code>\oddsidemargin = 62pt</code> | 4 | <code>\topmargin = 20pt</code> |
| 5 | <code>\headheight = 12pt</code> | 6 | <code>\headsep = 25pt</code> |
| 7 | <code>\textheight = 592pt</code> | 8 | <code>\textwidth = 327pt</code> |
| 9 | <code>\marginparsep = 10pt</code> | 10 | <code>\marginparwidth = 106pt</code> |
| 11 | <code>\footskip = 30pt</code> | | <code>\marginparpush = 5pt (not shown)</code> |
| | <code>\hoffset = 0pt</code> | | <code>\voffset = 0pt</code> |
| | <code>\paperwidth = 597pt</code> | | <code>\paperheight = 845pt</code> |

図 8.1 版面のレイアウトに使用できる長さ

`\voffset` 横組みにおいて用紙の左上の部分に入れる縦方向の余白。この値を 0 にしてもすでに 1 インチ分の余白が挿入されています。本当に用紙の左上端から使うならば `\voffset` を `'-1in'` に設定します。

`\hoffset` 横組みにおいて用紙の左上の部分に入れる横方向の余白。縦方向と同じようにすでに 1 インチ分の余白が挿入されています。

`\oddsidemargin` ページが奇数のときに挿入される左側の余白。文書クラスオプションに `oneside` を使っていると全てのページに `\oddsidemargin` が挿入されます。

`\evensidemargin` ページが偶数のときに挿入される左側の余白。文書クラスオプションに `twoside` を使っているときだけ有効で `oneside` では意味がありません。

ヘッダの設定に関する長さです。

`\topmargin` `\voffset` とヘッダの間隔です。

`\headheight` ヘッダの高さです。

`\headsep` ヘッダと本文領域の間隔です。

`\footskip` フッタ下部と本文領域の最下部との間隔です。

本文領域や傍注領域に関わる長さです。

`\textheight` 本文領域の高さです。ヘッダやフッタの高さは含まれません。

`\textwidth` 本文領域の幅です。

`\marginparwidth` 傍注の幅です。

`\marginparpush` 傍注と傍注のあいだの縦方向の長さです。

`\marginparsep` 傍注と本文領域との間隔です。

`\columnsep` 2 段組以上での段と段の間隔です。

`\columnseprule` 2 段組以上での段と段のあいだに入る罫線です。

通常ここで紹介した長さはクラスファイル側でフォントサイズやクラスオプションに応じて適切に設定されますので徒に変更しないでください。相手先の都合で「1 行何文字 1 ページ何行」のような設定などをしなければならないときは無理やり次のようにすることもできます。

```
\setlength{\textwidth}{33zw}
\setlength{\textheight}{40\baselineskip}
```

▼ 8.1.2 簡単なページレイアウト——geometry

例えば、「上下左右の余白を 2 cm とし、残りの領域は本文に使い、フッターとヘッダーは本文の領域の高さに含める」というページレイアウトにしたければ、大雑把には次のような設定をする事になります。

```
\voffset = -0.54 cm
\hoffset = -0.54 cm
\evensidemargin = 0 cm
\oddsidemargin = 0 cm
\textheight = \paperheight - 4 cm - \topmargin
               - \headheight - \headsep - \footskip
\textwidth = \paperwidth - 4 cm
```

これにより、L^AT_EX では次のような設定をする事になります。

```
\setlength \voffset {-1in}% \voffset の設定
\addtolength \voffset {2cm}
\setlength \hoffset {-1in}% \hoffset の設定
\addtolength \hoffset {2cm}
\setlength \textheight {\paperheight}% \textheight の設定
\addtolength \textheight {-4cm}
\addtolength \textheight {-\topmargin}
\addtolength \textheight {-\headheight}
\addtolength \textheight {-\headsep}
\addtolength \textheight {-\footskip}
\setlength \textwidth {\paperwidth}% \textwidth の設定
\addtolength \textwidth {-4cm}
\setlength \evensidemargin {0pt}% 偶数ページマージン
\setlength \oddsidemargin {\evensidemargin}% 奇数ページマージン
\setlength \fullwidth {\textwidth}
```

しかし、両面印刷をするとか、ヘッダーやフッターの高さを含む等の調整が絡んでくると、少々入り組んだ設定になってしまいます。そこで、もう少し簡単に版面の設定をしたいならば梅木秀雄氏の作成した geometry を使うのが良いでしょう。geometry パッケージは非常に多機能で、その全てを本書で紹介する事はできませんが、初学者がもっとも苦勞し、様々な問題に直面する事が多いように見受けられますので、なるべく詳細に解説します。

使い方は非常に簡単で、プリアンブルで geometry パッケージをオプション付きで読み込むだけです。

```
\usepackage[margin=2cm]{geometry}
```

上記のようにすると文章領域の上下左右の余白を 2 cm に設定します*1。

他には `geometry` パッケージを読み込んだ後に `\geometry` コマンドを使う方法です。これはプリアンブルのみで使う事ができます。

```
\usepackage[a5paper]{geometry}
\geometry{hmargin={3cm,0.8in},height=8in}
\geometry{height=10in}
```

`geometry` パッケージは `calc` パッケージにも対応していますので、次のような記述も可能です。

```
\usepackage{calc}
\usepackage[tehtheight=20\baselineskip+10pt]{geometry}
```

`jsclasses` に含まれる、`jsbook` クラスを用いている場合は `\fullwidth` を `\textwidth` に設定するのが良い時があります。

```
\setlength \fullwidth{\textwidth}
```

以下に `geometry` のパッケージオプションを挙げます。パッケージオプションは基本的には `\geometry` 命令の中で使うことができます。

■用紙サイズ

既定のサイズ `a0paper`, `a1paper`, `a2paper`, `a3paper`, `a4paper`, `a5paper`, `a6paper`, `b0paper`, `b1paper`, `b2paper`, `b3paper`, `b4paper`, `b5paper`, `b6paper`, `letterpaper`, `executivepaper`, `legalpaper`, `screen`. `screen` は 225 mm × 180 mm になります。 `screen` は `centering` も併用すると便利です。 '`paper=<用紙サイズ>`' と記述しても大丈夫です。

paperwidth 用紙の幅を、〈長さ〉を指定して決めます。 '`paperwidth=10cm`' のように使います。

paperheight 用紙の高さを、〈長さ〉を指定して決めます。

papersize '`papersize={〈幅〉,〈高さ〉}`' とするか、 '`papersize=<長さ>`' とすれば `paperwidth` と `paperheight` を用いた事と等価になります。

*1 用紙にはヘッダ、フッタ、傍注がありますから、これらの領域を除いた文面の余白が 2 cm という事になります。

landscape 横置きでページレイアウトを設定します。

portrait 縦置きでページレイアウトを設定します。

■本文のパラメータ

hscale 用紙の幅 (`\paperwidth`) に対して本文領域が占める横幅の比率です。‘`hscale=.8`’ とすると ‘`width=.8\paperwidth`’ と等価になります。標準は 0.7 です。

vscale 用紙の高さ (`\paperheight`) に対して本文領域が占める高さの比率です。‘`vscale=.8`’ とすると ‘`height=.9\paperheight`’ と等価になります。標準は 0.7 です。

scale 本文が幅と高さに関して用紙に対して占める比率を指定します。‘`scale={〈幅の比率〉,〈高さの比率〉}`’ とするか ‘`scale=〈比率〉`’ として使います。標準は 0.7 です。

width/totalwidth 本文のトータルな幅を指定します。‘`width=〈長さ〉`’ とするか、‘`totalwidth=〈長さ〉`’ とします。

height/totalheight 本文のトータルな高さを指定します。‘`height=〈長さ〉`’ とするか、‘`totalheight=〈長さ〉`’ とします。

total 本文のトータルな幅と高さを指定します。‘`total={〈幅〉,〈高さ〉}`’ とするか、‘`total=〈長さ〉`’ とします。

textwidth 文章領域となる `\textwidth` の幅を指定します。‘`textwidth=〈長さ〉`’ とします。

textheight 文章領域となる `\textheight` の高さを指定します。‘`textheight=〈長さ〉`’ とします。

text/body `\textwidth` と `\textheight` の両方を指定します。‘`body={〈幅〉,〈高さ〉}`’ とするか ‘`text=〈長さ〉`’ とします。

lines `\textheight` を行数によって決めます。‘`lines=〈整数〉`’ で指定します。

includehead トータルな本文の高さ `height/totalheight` にヘッダー (`\headheight` と `\headsep`) を含めるようにします。標準では無効です。

includefoot トータルな本文の高さ `height/totalheight` にフッター (`\footskip`) を含めるようにします。標準では無効です。

includeheadfoot `includehead` と `includefoot` の両方を有効にします。

includemp トータルな本文の幅に傍注 (`\marginparwidth` と `\marginparsep`) も

含めるようにします。 `marginparwidth` と `marginparsep` オプションに依存しています。標準では無効です。

`includeall` `includeheadfoot` と `includemp` の両方を指定した事と等価です。

`ignorehead` トータルな本文の高さにヘッダーを含めないようにします。標準で有効です。 '`includehead=false`' とするのと等価です。

`ignorefoot` トータルな本文の高さにフッターを含めないようにします。標準で有効です。 '`includefoot=false`' とするのと等価です。

`ignoreheadfoot` `ignorehead` と `ignorefoot` の両方を指定した事と等価です。

`ignoremp` トータルな本文の幅に傍注を含めないようにします。標準で有効です。

`ignoreall` `ignoreheadfoot` と `ignoremp` の両方を指定した事と等価です。

`heightrounded` 本文の高さが行送りの倍数でない場合に、“`underfull vbox`”の警告を出さないように `\textheight` を $\langle \text{baselineskip} \text{の整数倍} + \text{topskip} \rangle$ にします。

`hdivide` 左余白, 文章の幅, 右余白を指定します。 '`hdivide={⟨左余白⟩,⟨文章幅⟩,⟨右余白⟩}`' のように使います。このオプションは三つのうち二つだけ明確な時に、不定の一つを星 '*' に置き換えて '`hdivide={2cm,15cm,*}`' とできます。

`vdivide` 上余白, 文章の高さ, 下余白を指定します。 '`vdivide={⟨上余白⟩,⟨文章幅⟩,⟨下余白⟩}`' のように使います。

`divide` '`divide={⟨長さ1⟩,⟨長さ2⟩,⟨長さ3⟩}`' とすると '`hdivide={⟨長さ1⟩,⟨長さ2⟩,⟨長さ3⟩}`' と '`vdivide={⟨長さ1⟩,⟨長さ2⟩,⟨長さ3⟩}`' を指定した事と等価になります。

■余白

`left/lmargin/inner` 用紙の左端と本文領域(版面)とのあいだにある左余白を指定します。 '`lmargin=⟨長さ⟩`' のように使います。両面印刷指定 (`twoside`) の場合はノドの長さを設定します。

`right/rmargin/outer` 用紙の右端と本文領域とのあいだにある右余白を指定します。 '`rmargin=⟨長さ⟩`' のように使います。両面印刷指定 (`twoside`) の場合は小口の長さを設定します。

`top/tmargin` 用紙の上端と本文領域のとのあいだ(天)にある上余白を指定します。 '`tmargin=⟨長さ⟩`' のように使います。

`bottom/bmargin` 用紙の下端と本文領域のとのあいだ(地)にある下余白を指定し

ます. ‘`bmargin=⟨長さ⟩`’ のように使います.

hmargin 左右余白を指定します. ‘`hmargin={⟨左余白⟩,⟨右余白⟩}`’ とするか, ‘`hmargin=⟨長さ⟩`’ とします.

vmargin 上下余白を指定します. ‘`vmargin={⟨上余白⟩,⟨下余白⟩}`’ とするか, ‘`vmargin=⟨長さ⟩`’ とします.

margin ‘`margin={⟨長さ1⟩,⟨長さ2⟩}`’ とすると, ‘`hmargin={⟨長さ1⟩,⟨長さ2⟩}`’ と ‘`vmargin={⟨長さ1⟩,⟨長さ2⟩}`’ を指定した事と等価になります.

hmarginratio 左右余白の比率を指定します. ‘`hmarginratio=⟨左の比率:右の比率⟩`’ のようにコロンで区切ります. 正の整数値で 100 以下である必要があります. 片面印刷時 (*oneside*) は 1 : 1, 両面印刷時 (*twoside*) は 2 : 3 が標準です.

vmarginratio 用紙の上余白と下余白 (天地) の比率を指定します.

marginratio ‘`marginratio={⟨左右の比率⟩,⟨上下の比率⟩}`’ とするか, ‘`marginratio=⟨比率⟩`’ とします.

hcentering ‘`hmarginratio=1:1`’ とした事と等価になります.

vcentering ‘`vmarginratio=1:1`’ とした事と等価になります.

centering ‘`marginratio=1:1`’ とした事と等価になります.

twoside 両面印刷時に文章領域 (版面) が左右対称になるようにします.

asymmetric 両面印刷時に文章領域が左右非対称になるようにします. ただし, `bindingoffset` は考慮されます.

bindingoffset 片面印刷時でも両面印刷時においても文書を綴じる背の部分の余白を指定します. ‘`bindingoffset=⟨長さ⟩`’ として使います.

■レイアウトパラメータの設定 L^AT_EX のページレイアウトにおけるパラメータを設定する事もできます.

headheight/head ‘`head=⟨長さ⟩`’ として `\headheight` を設定します.

headsep ‘`headsep=⟨長さ⟩`’ として `\headsep` を設定します.

footskip/foot ‘`foot=⟨長さ⟩`’ として `\footskip` を設定します.

nohead ヘッダー無しの設定にします (`\headheight` と `\headsep` を 0pt にします).

nofoot フッター無しの設定にします (`\footskip` を 0pt にします).

noheadfoot `nohead` と `nofoot` を指定した事と等価になります.

- marginparwidth/marginpar*** ‘marginpar=〈長さ〉’として傍注の幅 `\marginparwidth` を設定します。
- marginparsep*** ‘marginparsep=〈長さ〉’として傍注と本文の空き `\marginparsep` を設定します。
- nomarginpar*** 傍注無しの設定にします (`\marginparwidth` と `\marginparsep` を 0pt にします)。
- columnsep*** ‘columnsep=〈長さ〉’として段間の空き `\columnsep` を設定します。
- hoffset*** ‘hoffset=〈長さ〉’として `\hoffset` を設定します。
- voffset*** ‘voffset=〈長さ〉’として `\voffset` を設定します。
- offset*** ‘offset={〈水平方向のオフセット量〉,〈垂直方向のオフセット量〉}’とするか ‘offset=〈オフセット量〉’として、オフセットを指定します。
- twocolumn*** 2段組の設定にします。
- twoside*** 両面印刷用に傍注の位置を入れ換え、紙面を左右対称になるようにします。
- textwidth*** ‘textwidth=〈長さ〉’として `\textwidth` を直接指定します。
- textheight*** ‘textheight=〈長さ〉’として `\textheight` を直接指定します。
- reversemp/reversemarginpar*** 傍注の位置を非標準の左側に出力するようにします。両面印刷時は見開きの内側 (小口) に設定します。

■デバイスドライバ

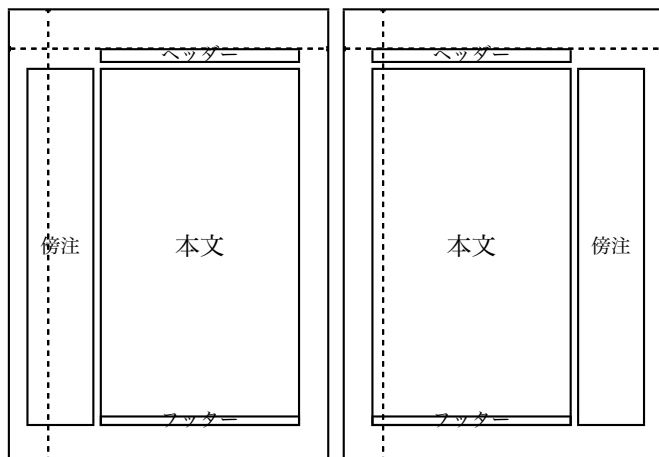
- dvips*** dvips 用に用紙サイズ情報を設定します。
- dvipdfm*** Dvipdfm 用に用紙サイズ情報を設定します。
- pdftex*** pdf \TeX , pdf \LaTeX 用に用紙サイズ情報を設定します。

■その他

- verbose*** geometry パッケージの処理内容を詳細に表示します。
- showframe*** ページレイアウトの設定を確認するために、1 ページ目の本文領域、ヘッダー、フッターを枠で囲みます。

jsbook クラスでのレイアウト例を示します。まずパッケージを読み込まない状態でのレイアウトです。

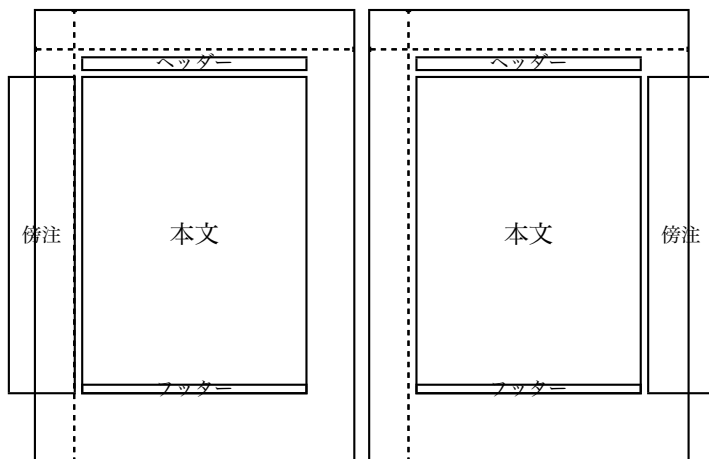
パッケージ無し,



▶ **問題 8.2** 上下左右の余白をちょうど 2cm に設定するにはどうすれば良いか考えてください。この場合はヘッダー、フッター、傍注の領域が含まれない事も確認してください。

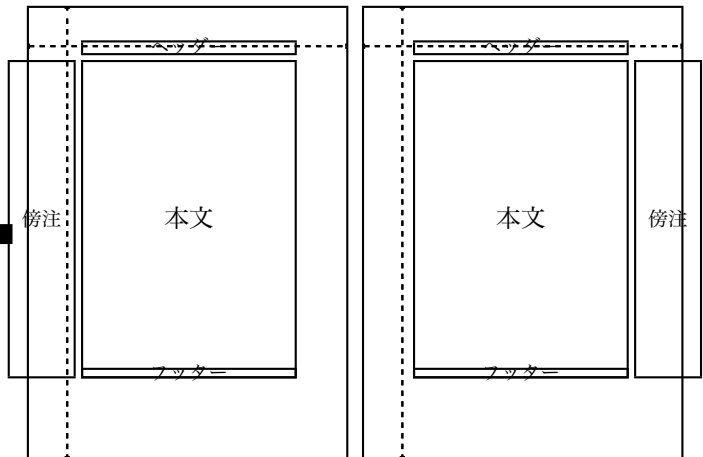
次は上下の余白の比率を 1:1, 左右の余白の比率を 1:1 に指定した例です。

centering,



この場合もヘッダー、フッター、傍注の領域は含まれません。次は見開きの内側の余白を 1cm 足した例です。

twoside,
bindingoffset=1cm;



▶ **問題 8.3** 左余白を 3cm, 右余白を 2cm, 行数を 40 行送り分, 上余白を 2.5cm として, ヘッダーとフッターの領域をトータルな高さを含めるような geometry の設定を考えてください. いくつも解答はありますが, 例えば次のような設定の実行結果を吟味してください.

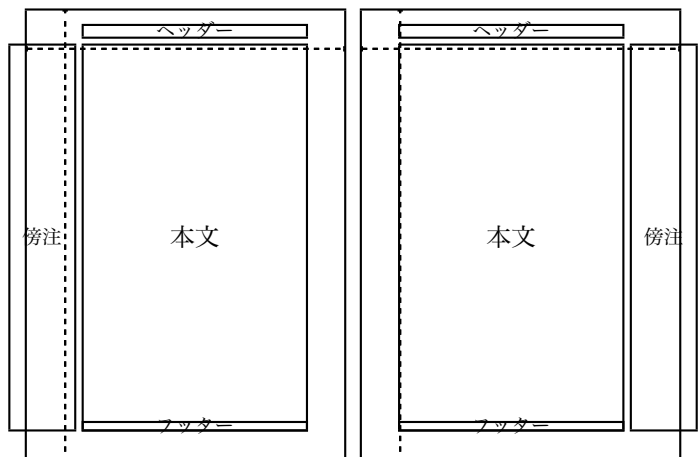
```
\geometry{left=3cm,right=2cm,lines=40,top=2.5cm,includeheadfoot}
```

また, これは次のようにしても同じである事を確認してください.

```
\geometry{hmargin={3cm,2cm},tmargin=2.5cm,lines=40,includeheadfoot}
```

次は本文の高さを 10 inch, 下余白を 2cm, 残りは上余白にするような設定です.

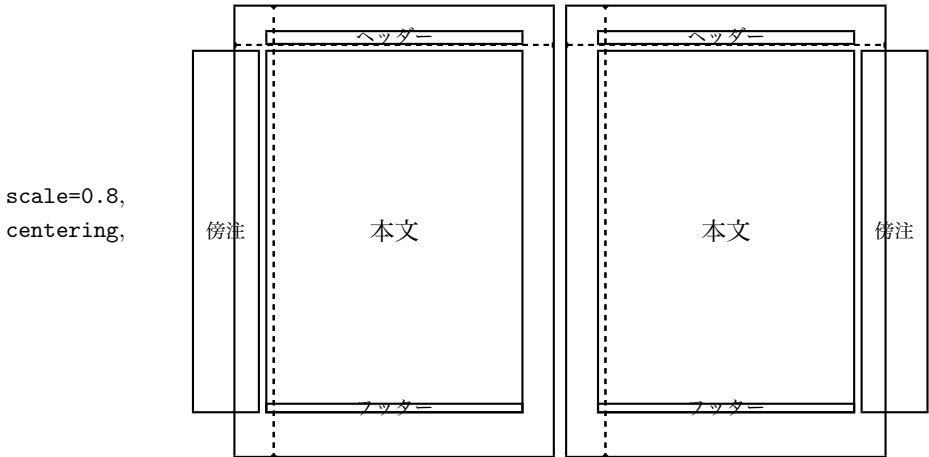
vdivide=
{*,10in,2cm},



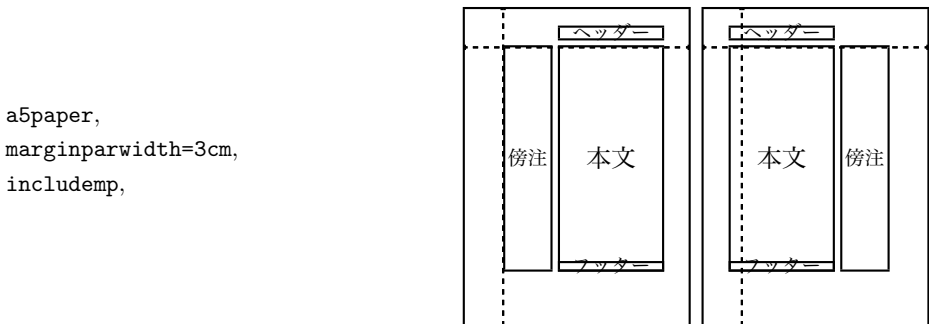
これは次のようにしても同じ事になります。

```
\geometry{bottom=2cm,textheight=10in}
```

次は幅も高さも用紙の 80% だけ本文の領域に割り当てるようにし、用紙の中央に本文が配置するようにします。

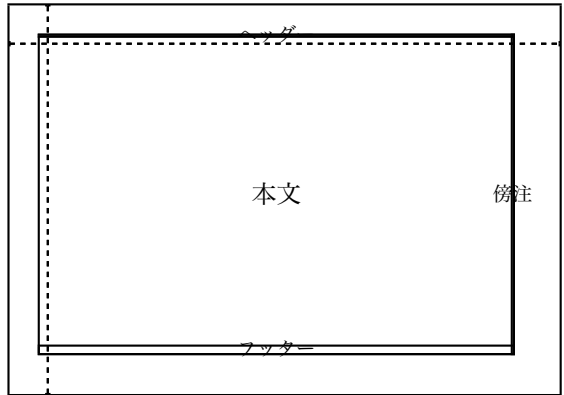


用紙サイズを A5 とし、傍注の幅を 3 cm、傍注の幅もトータルな幅に含めるようにします。



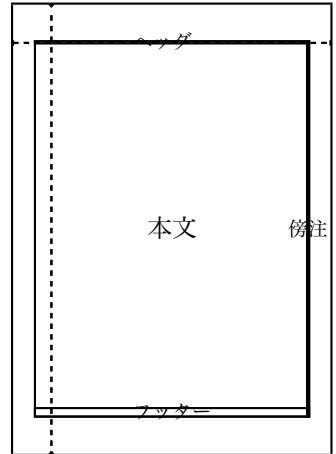
次は用紙サイズを B4 とし、横置き、2 段組、左右上下の余白は 2 cm、傍注・ヘッダー・フッターなしで、段間の幅を 2 cm とします。

```
dvipdfm,
b4paper,
landscape,
twocolumn,
margin=2cm,
nomarginpar,
nofoot,
nohead,
columnsep=2zw,
```



次は用紙サイズを A4 として、2 段組、左右余白をそれぞれ 15 mm，上下余白を 25 mm，段間の幅を 6 mm，傍注・ヘッダー・フッターなしの設定にします。

```
a4paper,
twocolumn,
ignoreall,
nomarginpar,
noheadfoot,
margin={15mm,25mm},
columnsep=6mm,
```



▼ 8.1.3 ヘッダやフッタの設定その 1

ヘッダやフッタなどに出力されるページ番号などを変更したいときがあると思います。

```
\pagestyle{<表示形式>}
```

`\pagestyle` 命令を記述したページから指定した表示形式に変更されます。指定できる形式は表 8.1 の通りです。

‘`myheadings`’ では二つの命令によってヘッダの出力を指定します。

表 8.1 ヘッダやフッタの指定

命令	内容
<code>empty</code>	ページ番号を表示しない
<code>plain</code>	フッタ中央部に表示する
<code>headings</code>	ヘッダにページ番号と章・節名を表示する
<code>myheadings</code>	ユーザ定義の表示形式にする

```
\markright {<ヘッダ>}
\markboth {<偶数ヘッダ>}{<奇数ヘッダ>}
```

片面印刷のときに `\markright` を使います。両面印刷には `\markboth` を使います。2006 年度版の「〇〇△△大学」の論文集を作成しているのであれば、例えば次のようにします。

```
\pagestyle{myheadings}
\markboth{〇〇△△大学論文集}{2006年度版}
```

任意の 1 ページだけのヘッダ・フッタは、次のようにする事でそのページだけ変えられます。

```
\thispagestyle{<表示形式>}
```

ページ番号の表示形式を変更するには `\pagenumbering` 命令を使います。

```
\pagenumbering{<表示形式>}
```

その場所から指定した形式で 1 ページ目からカウントしてページ番号を表示します。指定できる表示形式は表 8.2 の通りです。

ここで注意する事はアルファベットにした場合は、最大 26 ページまでしかカウントできないという事です。27 以上になった場合の対策は別にする事になります。

▼ 8.1.4 ヘッダ・フッタの変更その 2——`fancyhdr`

L^AT_EX が標準で用意してくれているページスタイルでは寂しい、そう思う人も多いでしょう。自分で全て定義する事もできますが、Piet Oostrum 氏が作成した `fancyhdr` を使うと比較的簡単にページスタイルをカスタマイズできます。`fancyhdr` は `fancyheadings`

表 8.2 ページ番号の種類指定

形式	内容	出力例
arabic	アラビア数字	1, 2, 3, ...
roan	ローマ数字	i, ii, iii, ...
Roman	ローマ数字	I, II, III, ...
alph	アルファベット小文字	a, b, c, ..., z
Alph	アルファベット大文字	A, B, C, ..., Z

の後継で、ページのヘッダーとフッターをカスタマイズできるマクロです。まず `fancyhdr` を使うために以下をプリアンブルに記述します。

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

`fancyhdr` ではヘッダ・フッタを六つに分割しています。

<code>\lhead</code>	<code>\chead</code>	<code>\rhead</code>
(<code>\headrulewidth</code>)		
(<code>\footrulewidth</code>)		
<code>\lfoot</code>	<code>\cfoot</code>	<code>\rfoot</code>

`\lhead`, `\chead`, `\rhead` の三つはヘッダに使い `\lfoot`, `\cfoot`, `\rfoot` の三つはフッタに使います。 `\headrulewidth` はヘッダ下部の罫線の太さで、 `\footrulewidth` はフッタ上部の罫線の太さです。図だけのページや表だけのページはシンプルなヘッダ・フッタにしたり、ヘッダ下部の罫線を引かない場合があります。その場合は `\headrulewidth` を `\iffloatpage` 命令で判定するように定義すると良いでしょう。

```
\def\headrulewidth{\iffloatpage{0pt}{.4pt}}
```

`\lhead` などの他の命令も同様に変更できます。

例えば `jarticle` クラスで次のようなヘッダ・フッタ

資本主義社会の崩壊	
(<code>\textwidth</code>)	
名無しの権兵衛	ページ番号

を設定したければ以下のように記述します。

```

\lhead{} \chead{}
\rhead{資本主義社会の崩壊}
\lfoot{名無しの権兵衛}
\cfoot{}
\rfoot{\textbf{\thepage}}
\def\headrulewidth{.4pt}
\def\footrulewidth{.4pt}

```

書籍用クラス `jbook` などクラスオプションに `twoside` が指定されている場合は偶数ページと奇数ページを個々に設定します。例えば次のようなヘッダ・フッタにしたいとします。

第 1 章 序論	1.1 背景
本文	
2	
1.2 目標	第 1 章 序論
本文	
3	

ヘッダ・フッタの奇数ページと偶数ページの場所を区別するために以下のような設定になっています。

EL(H)	EC(H)	ER(H)
本文		
EL(F)	EC(F)	ER(F)
偶数ページ側		
OL(H)	OC(H)	OR(H)
本文		
OL(F)	OC(F)	OR(F)
奇数ページ側		

先程の出力を得るためには `\fancyhead` と `\fancyfoot` 命令を使います。

```

\documentclass{jbook}
\usepackage{fancyhdr} \pagestyle{fancy}
\fancyhead[ER,OL]{\rightmark}
\fancyhead[EL,OR]{\leftmark}
\fancyfoot[EC,OC]{\textbf{\thepage}}

```

欧文のクラスではこれで良いのですが和文では `\chaptermark` と `\sectionmark` の定義を `fancyhdr` を読み込んだ後に次のように定義するのが普通でしょう。

```

\def\chaptermark#1{\markboth{%
  \ifnum \c@secnumdepth >\m@ne
    \if@mainmatter
      \@chapapp\thechapter\@chappos\hskip1zw
    \fi
  \fi
  #1}{}}%
\def\sectionmark#1{\markright{%
  \ifnum \c@secnumdepth >\z@ \thesection \hskip1zw\fi
  #1}}%

```

▼ 8.1.5 ページ/総ページ

ヘッダ・フッタの設定をフッタだけに「ページ/総ページ」にしたい場合があります。これは例えば次のようにプリアンブルに記述します。

```

\AtEndDocument{\label{lastpage}}
\makeatletter
\newcommand{\ps@total}{%
\let\@mkboth\@gobbletwo
\let\@oddhead\@empty
\let\@evenhead\@empty
\def\@oddfont{\normalfont\hfil--\thepage/\pageref{lastpage}--\hfil}%
\let\@evenfoot\@oddfont}
\makeatother
\pagestyle{total}

```

この場合は `\ps@total` によって新規に ‘total’ というページスタイルを定義しています。ページ番号の書体を変えたいときは `\normalfont` を `\bfseries` などに変更します。

8.2 レイアウトの制御

LaTeX ではユーザ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。

`\samepage` 指定した場所でできる限り改ページを抑制します。

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。

これらの空白制御の命令では単位付きの長さで指定します。

`\hspace{1cm}` 空白制御用のコマンドは行頭では意図的に `\vspace{1cm}` 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。

8.3 その他のコマンド

L^AT_EX で用意されているその他のコマンドを紹介します。

▼ 8.3.1 日付

L^AT_EX のプログラムを実行した段階で、その原稿をタイプセットした日付を保存しています。 `\hour` と `\minute` は `jclasses/jsclasses` でのみ使えます。

<code>\today</code>	(日付)	<code>\month</code>	(月)	<code>\hour</code>	(時)
<code>\year</code>	(年)	<code>\day</code>	(日)	<code>\minute</code>	(分)

使用しているクラスファイルによって出力が違います。 `jarticle` においては `\西暦` や、 `\和暦` という命令を使って `\today` の西暦表示と和暦表示を変更できます。奥村晴彦

氏の `jsclasses` で標準は西暦になっており、アスキーの `jclasses` では和暦が標準です（個人的には天皇制の名残のような和暦を使うのは好ましくないと感じていますし、ビジネス文書でわざわざ和暦を使っても世界に置いてけぼりを食らうだけだと思っています、個人的に）。欧文のクラスファイルではその言語の標準的な表示方法で出力されます。`\today` 以外は `\number` 等のカウンタの値を表示するコマンドを必要とします*2。

今日は `\number\year` 年 `\number\month` 月 `\number\day` 日です。略して `\today`。
今日は 2016 年 7 月 23 日です。略して 2016 年 7 月 23 日。

▼ 8.3.2 L^AT_EX のロゴ

`\TeX` (T_EX) `\LaTeX` (L^AT_EX) `\LaTeXe` (L^AT_EX 2_ε)

奥村晴彦氏の `jsclasses` ではこれらに加えて次のロゴが用意されています。

`\pTeX` (pT_EX) `\pLaTeX` (pL^AT_EX) `\pLaTeXe` (pL^AT_EX 2_ε)

「実は僕も `{\TeX}` 使っています。」 \\
「え？ `{\LaTeX}` じゃないんですか？」 \\
「いやあ、僕は `{\TeX pert}` だからさ、
`{\LaTeXe}` も使ってませんよ。」 \\
「ということは `{\pTeX}` は使うんですね？」

「実は僕も T_EX 使っています。」
「え？ L^AT_EX じゃないんですか？」
「いやあ、僕は T_EXpert だからさ。L^AT_EX 2_ε
も使ってませんよ。」
「ということは pT_EX は使うんですね？」

8.4 単位・通貨の出力について

文中でも数式中でも単位は基本的にはローマン体で出力するのが普通ですので

それは `$y=30cm$` だから合計 300mm だ。

それは `y = 30cm` だから合計 300mm だ。

のような入力はおかしい訳です。この場合は

それは `$y=30\,\mathrm{cm}$` だから
合計 300\,mm だ。

それは `y = 30 cm` だから合計 300 mm だ。

としたほうが良いでしょう*3。このように単位は数式中でも使う事があるかもしれませ

*2 `\two@digits` 命令を用いると '2006/04/15' のように、ゼロが補われるようになります。

*3 場合によっては '`y = 30 [cm]`' とする事もあると思います。

るので `\ensuremath` で単位用の命令を作成します。例えば長さの単位である ‘mm’ は次のように定義します。

```
\newcommand{\mm}{\ensuremath{\,\,\mathrm{mm}}}
```

ただし L^AT_EX ですでに定義されている短い命令は再定義しないほうが無難ですので、リットル l などは次のようにします。

```
\newcommand{\litter}{\,\,\ensuremath{\mathrm{1}}}
```

日本ではリットルをイタリック体にする事もあるようですが、基本的に単位は全てローマン体にしても良いようです。分野によっても区別が違うので調べてください。毎回単位を定義するのも大変なので単位用のマクロ `units.sty` を以下のように作成します。

```
%File:units.sty
\newcommand{\mm}{\ensuremath{\,\,\mathrm{\milli mm}}}
\newcommand{\cm}{\ensuremath{\,\,\mathrm{cm}}}
\newcommand{\Km}{\ensuremath{\,\,\mathrm{Km}}}
\newcommand{\mg}{\ensuremath{\,\,\mathrm{mg}}}
\newcommand{\Kg}{\ensuremath{\,\,\mathrm{Kg}}}
\newcommand{\cc}{\ensuremath{\,\,\mathrm{cc}}}
\newcommand{\litter}{\,\,\ensuremath{1}}
\newcommand{\Ohm}{\ensuremath{\,\,\mathrm{\Omega}}}
```

「あの単位の命令はなんだったかなあ」と考えているよりも `\mathrm` などを使ったほうが早いかもしれません。

▶ **問題 8.4** `units.sty` はかなり汎用性に欠ける書き方です。これをどのような単位でも対応できるように一般化してください。

例えば ‘\U’ という単位を意味する命令を新規に定義するというならば、おおむね次のようになります。

```
\newcommand*\U[1]{\ensuremath{\,\,\mathrm{#1}}}
\newcommand*\BU[1]{\ensuremath{\,\, [\mathrm{#1}]}}
```

さらに ‘ $x = 1 \text{ [cm]}$ ’ という出力を ‘ $x = 1 \text{ \BU{cm}}$ ’ として括弧付きで用いるようにするのも便利でしょう。

通貨などを出力するためには L^AT_EX に標準で含まれる `textcomp` パッケージを使うと良いでしょう。これは古いエンコーディングだと使えませんが `fontenc` パッケージを読み込み次のようにします。

```
\usepackage[T1]{fontenc}
\usepackage{textcomp}
```

フォントがビットマップになるという事も危惧されますので特に不都合がなければ `txfonts` や `pxfonts` を併用して

```
\usepackage{txfonts,textcomp}%とかpxfonts
```

とするのがベターだと思います。表 8.3 が `textcomp` によって使用できる記号一覧です。`textcomp` にも含まれていない通貨などを探しているときは CTAN に記号の見本*4がありますのでそちらを参照してください。

8.5 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 8.4 の見出しが定義されています。

`\bibname` 命令は `jreport` や `jbook` などで定義で (j)article では `\refname` となっています。奥村晴彦氏の `jsclasses` 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}
```

上記のように `\presectionname` や `\postsectionname` を再定義します。

8.6 目次再び

目次に出力される項目を制御したいときがあります。例えば見出し命令にアスタリスクをつけた場合 (`\chapter*` など) は通し番号が付かずに目次にも出力されません。これを目次にも書き出すには次のようにします。

```
\addcontentsline{toc}{chapter}{なんとか}
```

*4 CTAN: [info/symbols/comprehensive/](http://info.symbols/comprehensive/)

表 8.3 textcomp で使える記号

— \texttwelveudash	~ \texttildelow	} \textrightarrow
← \textleftarrow	= \textdblhyphenchar	¢ \textcent
→ \textrightarrow	˘ \textasciibreve	£ \textsterling
␣ \textblank	ˇ \textasciicaron	¤ \textcurrency
\$ \textdollar	“ \textgravedbl	¥ \textyen
' \textquotesingle	” \textacutedbl	\textbrokenbar
= \textdblhyphen	† \textdagger	§ \textsection
o \textzerooldstyle	‡ \textdaggerdbl	¨ \textasciidieresis
1 \textoneoldstyle	‖ \textbardbl	© \textcopyright
2 \texttwooldstyle	‰ \textperthousand	ª \textordfeminine
3 \textthreeoldstyle	• \textbullet	◊ \textcopyleft
4 \textfouroldstyle	°C \textcelsius	¬ \textlnot
5 \textfiveoldstyle	\$ \textdollaroldstyle	Ⓟ \textcircledP
6 \textsixoldstyle	¢ \textcentoldstyle	® \textregistered
7 \textsevenoldstyle	f \textflorin	— \textasciimacron
8 \texteightoldstyle	Ⓒ \textcolonmonetary	° \textdegree
9 \textnineoldstyle	₩ \textwon	± \textpm
< \textlangle	₪ \textnaira	² \texttwosuperior
- \textminus	₲ \textguarani	³ \textthreesuperior
> \textrangle	₱ \textpeso	´ \textasciiacute
Ⓜ \textmho	₳ \textlira	μ \textmu
○ \textbigcircle	₴ \textrecipe	¶ \textparagraph
Ω \textohm	‡ \textinterrobang	· \textperiodcentered
[\textlbrackdbl	‡ \textinterrobangdown	※ \textreferencemark
] \textrbrackdbl	đ \textdong	¹ \textonesuperior
↑ \textuparrow	™ \texttrademark	º \textordmasculine
↓ \textdownarrow	¶ \textpilcrow	√ \textsurd
` \textasciigrave	฿ \textbaht	¼ \textonequarter
* \textborn	№ \textnumero	½ \textonehalf
∅ \textdivorced	% \textdiscount	¾ \textthreequarters
† \textdied	€ \textestimated	€ \texteuro
♻ \textleaf	○ \textopenbullet	× \texttimes
∞ \textmarried	SM \textservicemark	÷ \textdiv
♪ \textmusicalnote	{ \textlquill	
	‰ \textpertenthousand	, \textquotestraightbase
	„ \textquotestraightdblbase	* \textasteriskcentered
	— \textthreequartersemdash	/ \textfractionsolidus

表 8.4 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>	章見出し番号の前の文字	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

```
\addcontentsline{<拡張子>}{<種類>}{<要素>}
\addtocontents{<拡張子>}{<要素>}
```

例えば文書クラスに `jreport` を使っていた場合に「謝辞」のような章を出力するときには次のように入力します。

```
\chapter*{謝辞}
ありがとう, 本当にありがとう.
```

しかしこれを目次にも追加するには `\chapter*` の直後に記述します。

```
\chapter*{謝辞}\addcontentsline{toc}{chapter}{謝辞}
ありがとう, 本当にありがとう.
```

目次や図目次のある部分で改ページしたいときには次のようにします。

```
\addtocontents{toc}{\newpage}
\addtocontents{lof}{\newpage}
```

8.7 多段組

LaTeX 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 段組みするには `\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ています。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。

▷ **例題 8.5** 次の入力例をタイプセットし、その出力結果を吟味して下さい。

```

\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 段組みにすると図表は用紙の文章幅 `\textwidth` ではなく 1 段分の幅 `\columnwidth` で張り込む事になります。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ています。

`\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。

▼ 8.7.1 3 段組以上の多段組——`multicol`

Frank Mittelbach 氏の作成した `multicol` を使うと最高で 10 段まで段組できます。自動的に段の終わりの最終ページの文章の高さを揃えてくれます。星を付けると段の下端を揃えないようにするので、通常は星無しで用いると良いでしょう。

```

\begin{multicols}{⟨段数⟩}
⟨文章内容⟩
\end{multicols}

```

`multicols` 環境の場合は改ページされずに同じページに違う段組を混在できます。余り好ましくない事なので多用しないほうが良いでしょう。以下のような入力があります。

```
\begin{multicols}{4}
このパッケージでは 10 段組まで多段組できますが、同ページに違う段数の要素
を組み込むのは余り好ましいことではないので、特別な理由がない限り使うべき
ではありません。このパッケージでは段の終わりの最終ページの文章を自動で揃
えるので、従来の組版の規則にも合っています。
\end{multicols}
```

そうすると、次のような出力を得ることができます。

このパッケージ	み込むのは余り好	ん。このパッケー	組版の規則にも合
では 10 段組まで	ましいことではな	ジでは段の終わり	っています。
多段組できます	いので、特別な理	の最終ページの	
が、同ページに違	由がない限り使う	文章を自動で揃	
う段数の要素を組	べきではありません	えるので、従来の	

LaTeX 2_ε の標準では `\onecolumn` と `\twocolumn` によって 1 段組みと 2 段組みを切替える事もできます。

これによって全体の段組みを指定する事もできます。しかし、`\onecolumn` と `\twocolumn` の両方とも改ページが必須で、ページの途中で段を変更する事もできなければ、2 段組みのときに段の終わりが揃わないなどの制約があります。

`multicol` パッケージの制約として `table` 環境や `figure` 環境で図表を出力する事ができません。そのかわり `table*` 環境並びに `figure*` 環境で段を打ち抜いて図表を出力する事ができます。この場合はページの上端か下端に配置されるようになります。余白等を手動で調整する手間を惜しまないならば、次の入力例のように自前の `mytable` 環境や `myfigure` 環境を定義する事ができます。

```
\documentclass[a4j,10pt,papersize]{jsarticle}
\newcommand*{\mypict}{\setlength\unitlength{1pt}%
\begin{picture}(40,40)%
\put(20,20){\circle*{10}}%
\put(20,20){\circle{20}}%
\end{picture}}%
\usepackage{multicol}
\columnseprule=.4pt
\columnsep=2zw
\multicolsep=1zw
```

```

\makeatletter
\def\myfigure{\vbox\bgroup\centering\def\@capttype{figure}}
\def\mytable{\vbox\bgroup\centering\def\@capttype{table}}
\def\endmyfigure{\egroup}
\def\endmytable{\egroup}
\def\hoge{\@tempcnta=\z@ \@whilenum \@tempcnta<100\do{%
  ほげほげ\advance\@tempcnta\@ne}。}
\makeatother
\begin{document}
\hoge
\begin{multicols}{3}
\hoge
\begin{equation}
f(x) = ax + b
\end{equation}
\hoge
\begin{figure*}[tb]
\centering\mypict
\caption{普通の\texttt{figure}環境での図だよ}
\end{figure*}
\hoge
\begin{myfigure}
\mypict
\caption{\texttt{multicols}環境中の図だよ}
\end{myfigure}
\hoge
\begin{mytable}
\begin{small}
\caption{\texttt{multicols}環境中の表だよ}
\begin{tabular}{lll}
\hline
\LaTeX & \LaTeXe & \LaTeX\,3\
\hline
\end{tabular}
\end{small}
\end{mytable}
\hoge
\end{multicols}
\hoge
\begin{multicols}{4}[\section{新聞記事とか色々あるけどねえ、どうだろう。}]
\hoge
\end{multicols}
\hoge
\begin{multicols}{5}
\hoge\par \hoge
\end{multicols}
\hoge

```

```
\begin{multicols}{7}
\hoge\par \hoge
\end{multicols}
\end{document}
```

上記の入力例の出力例は図 8.2 となります。

Frank Mittelbach 氏の multicols では段抜きで配置する figure* 環境と table* 環境は許されますが、段の中に組まれる figure/table 環境は使えません。そのため、図をフロートさせずに配置するために myfigure/mytable 環境を新設しています。ただし、フロートしないため図表の直前/直後で余計な空白が挿入されるので、これを手動で調整する (具体的には関係する文の少し後に myfigure/mytable を記述する) 事になります。

段間の空白、段間に引かれる罫線、段の上端/下端の空きは次のように設定します。

```
\columnseprule=.4pt % 段間に引かれる罫線
\columnsep=2zw % 段間の空白
\multicolsep=1zw % 段の上端/下端の空き
```

8.8 長さ

TeX/LaTeX における**長さ**には伸縮するものとししないもの 2 種類があります。伸縮するものを**可変長の長さ**と呼び、伸縮しないものを**固定長の長さ**と呼びます。可変長の長さは**スキップ**と呼ばれる事が多いようですが、本書では可変長の長さと呼びます。可変長の長さは**長さ**、**縮み率**、**伸び率**の三つの属性を持っています。固定長の長さは決まった値しか持ちません。可変長の長さは縮み率と伸び率に従ってバネのように伸縮します。長さの定義には `\newlength` が使えます。

```
\newlength{<綴り>}
\setlength{<綴り>}{<長さ>}
\addtolength{<綴り>}{<長さ>}
```

`\setlength` で長さを設定します。`\addtolength` では元の長さにさらに長さを足します。`\newlength` で定義された長さは可変長にも固定長にもなって良い事になっています。まず `\newlength` で新規に長さを定義します。次に `\setlength` で値を決めます。

<p>図 1 普通の <code>align</code> 環境での配列</p>	<p>図 2 <code>multicol</code> 環境での配列</p>
<p>図 3 <code>multicol</code> 環境での配列</p>	<p>図 4 <code>multicol</code> 環境での配列</p>

図 8.2 multicol の使用例の出力結果

<code>\newlength{\newa}\the\newa\</code>	0.0pt			
<code>\setlength{\newa}{10mm}\the\newa\</code>	28.45274pt			
<code>\setlength{\newa}{10mm plus 3mm minus 2mm}\the\newa\</code>	5.69054pt	plus	8.53581pt	minus
<code>\addtolength{\newa}{3mm}\the\newa</code>	36.98856pt	plus	8.53581pt	minus
	5.69054pt			

可変長の長さに値を足しても縮み率と伸び率には影響しないのが例の出力から分かるでしょう。

長さを設定するには次の命令も使えます。

```
\settowidth{<綴り>}{<要素>}
\settoheight{<綴り>}{<要素>}
\settodepth{<綴り>}{<要素>}
```

長さに対して要素の幅を代入するには `\settowidth` を使います。高さに関しては `\settoheight`、深さには `\settodepth` を使います。このような命令の使い道を少し紹介しておきます。

```
%\newlength{\newa}
\newcommand{\fakewidth}[1]{%
  \settowidth{\newa}{#1}%
  \framebox[\the\newa][c]{\strut}}
解は $\int f(x)dx$ となる。 \par
解は $\fakewidth{\int f(x)dx}$ となる。
```

解は $\int f(x)dx$ となる。

解は  となる。

8.9 箱の操作

まずは L^AT_EX 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。L^AT_EX 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。L^AT_EX の箱には**高さ**と**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。

▼ 8.9.1 枠のない箱

L^AT_EX ではなんとも簡単に複数の要素を一つの箱に収める事ができます。

```
\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’ の四つを使う事ができます。

```
\makebox[3zw][l]{未来}と
\makebox[3zw][c]{函館}と          未来 と 函館 と   北海道と G o o d !
\makebox[5zw][r]{北海道}と      です。
\makebox[5zw][s]{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
改行が起りません。
```

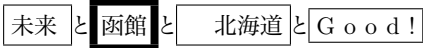
▼ 8.9.2 枠のある箱

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。箱には枠を付ける事もできます。

```
\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。
```



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`\fbox{びったりです}`、こちらは`\fbox{若干太い}`。

▼ 8.9.3 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、  
上が基準}\ldots  
\parbox[b]{3zw}[t]{4zw}{文字が下に、  
下が基準}\ldots
```

```
文字が下  
に、下が  
基準  
箱が中央  
に、…文字が中…  
央、上が  
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` 環境では段落が組まれますし、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}  
~となります。
```

```
この環境は ページを組みあ となります。  
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。

▼ 8.9.4 箱の保存と使用

ある要素を箱の中に保存し、それを再利用できればエネルギー消費を減らす事ができます。

```
\newsavebox{<綴り>}
```

箱を保存するためには保存する場所の確保を `\newsavebox` で行います。L^AT_EX が使っても良い箱は数が限られているのであらかじめいくつくらい使うのかを宣言してあげます。宣言するときはバックスラッシュを先頭に付けます。箱の中に要素を保存するときは `\savebox` か `\sbox` 命令を使います。

```
\savebox{<綴り>}[<幅>][<要素の位置>]{<要素>}
\sbox{<綴り>}{<要素>}
```

箱の幅や要素の位置を指定するときは `\savebox` を使います。もちろん幅を指定しないと要素の位置は指定できません。上記の命令のほかにも `lrbox` 環境があります。行に収まるくらいの要素を `lrbox` 環境の中に記述すると `<綴り>` の箱に代入します。

```
\begin{lrbox}{<綴り>}
<要素>
\end{lrbox}
```

これまでの命令では箱を用意してその中に要素を保存するだけですので、保存した箱を `\usebox` 命令で使います。

```
\usebox{<綴り>}
```

`\usebox` 命令を使うと `<綴り>` の箱に保存されている要素を複数回再利用できます。

<code>\newsavebox{\hoge}</code>	L ^A T _E X 2 _ε から L ^A T _E X 3 へ、
<code>\savebox{\hoge}{\LaTeXe から</code> <code>\LaTeX\,3 へ}</code>	L ^A T _E X 2 _ε から L ^A T _E X 3 へ、
<code>\usebox{\hoge}, \usebox{\hoge}.\.</code>	L ^A T _E X 2 _ε から L ^A T _E X 3 へ、
<code>\usebox{\hoge}, \usebox{\hoge}.\.</code>	L ^A T _E X 2 _ε から L ^A T _E X 3 へ。
<code>\sbox{\hoge}{ } \usebox{\hoge}</code>	

L^AT_EX の作業領域は限られていますので `\setbox` 命令で `<綴り>` の箱を使い終わったら空にします。

▼ 8.9.5 箱の上げ下げ

ある要素を箱の中に入れて、さらに上げ下げを同時に行う `\raisebox` 命令もあります。

```
\raisebox{<上げ下げ>}[<高さ>][<深さ>]{<要素>}
```

`\raisebox` 命令の中には文字列や他の箱も挿入できます。

それで、`\raisebox{1zw}{あれ}`は

`\raisebox{-1zw}{これ}`で、

`\raisebox{1.5zw}{\fbox{枠付きの箱}}`

だ。

それで、あれは、枠付きの箱で、これだ。

▼ 8.9.6 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事もできます。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、こんな使い方のできるわけです。枠の見える状態での例を見てください。

```
\newcommand*\RULE[2]{%
```

```
\rule{0pt}{#1}\rule{#2}{0pt}}
```

```
未来 \fbox{\RULE{3zw}{4zw}}
```

```
函館 \fbox{\RULE{3zw}{2zw}}
```

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` の中に箱を入れる事もできますし、何を入れても構いません。

`\underline{\fbox{枠付きの箱}}の下線`

はこのようにしますし、もちろん

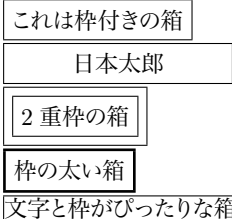
`\underline{下線}`も表示できます。

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。

▼ 8.9.7 枠付きの箱その 2——fancybox

L^AT_EX の標準では `\fbox` と `\framebox` 命令が使えます。

```
\fbox{これは枠付きの箱} \\
\framebox[10zw][c]{日本太郎} \\
\fbox{\fbox{2 重枠の箱}} \\
{\fboxrule=1pt\fbox{枠の太い箱}} \\
{\fboxsep=0pt\fbox{文字と枠がぴったり
な箱}}
```



Timothy Zandt 氏による fancybox を使ってみると枠付きの箱を出しやすいでしょう。枠付きの箱に文字列を入れるための命令として `\shadowbox`, `\doublebox`, `\ovalbox`, `\Ovalbox` の四つがあります。

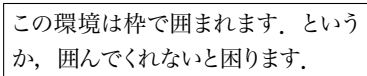
```
\shadowbox{影付きの箱} \\
\doublebox{2 重の枠} \\
\ovalbox{辺が丸い箱} \\
\Ovalbox{太くて辺が丸い箱}}
```



fancybox を使うと Sbox 環境というものが使えて、これは環境内のものを箱（レジスタ）である `\TheSbox` に保存します。このようにして保存した箱を `\fbox` などで囲みます。例えば `minipage` を枠で囲む `fminipage` 環境を作成するのであれば次のようにします。

```
\newenvironment{fminipage}%
  {\begin{Sbox}\begin{minipage}}%
  {\end{minipage}\end{Sbox}\fbox{\TheSbox}}
```

```
\newenvironment{fminipage}%
  {\begin{Sbox}\begin{minipage}}%
  {\end{minipage}\end{Sbox}}
  \fbox{\TheSbox}
\begin{fminipage}{.8\linewidth}
この環境は枠で囲まれます。というか、
囲んでくれないと困ります。
\end{fminipage}
```



数式環境などを枠で囲もうと思うとき、数式番号も含めて枠の中に入れるのはちょっと面倒かもしれません。`\fbox`などの命令を使うとそこから数式を組み立てるモードではなく文章を組み立てるモードになりますので、もう1度数式環境を書きます。

```
\( y = f(x) + \fbox{\(C\)})
\begin{equation}
  \fbox{\(y = f(x)\)}
\end{equation}
```

$$y = f(x) + \boxed{C}$$

$$\boxed{y = f(x)} \quad (8.1)$$

番号付きの数式を文章幅いっぱいに枠で囲むには

```
(\linewidth (文章幅) -2\fboxrule -2\fboxsep)
```

を計算します。これには `calc` パッケージが使えます。

```
\usepackage{calc,fancybox}
\fbox{\parbox{\linewidth-
  2\fboxrule-2\fboxsep}{
\begin{equation} y=f(x)
\end{equation}}}
```

$$\boxed{y = f(x)} \quad (8.2)$$

`eqnarray` 環境は枠で囲もうと思うと `fancybox` の場合は `Beqnarray` が用意されていますので、こちらの環境を `\fbox`などで囲みます。

```
\fbox{\begin{Beqnarray}
  y & = & f(x) \\
  & & = & 1/x
\end{Beqnarray}}
```

$$\boxed{\begin{array}{l} y = f(x) \\ = 1/x \end{array}} \quad \begin{array}{l} (8.3) \\ (8.4) \end{array}$$

`table` 環境や `figure` 環境などで見出しも含まれる要素は `\Sbox` で一度保存したものを枠で囲むようにします。こうすると

```
\usepackage{calc,fancybox}% プリアンブルで
\newenvironment{ftable}[1][htbp]%
  {\begin{table}[#1]
  \begin{Sbox}\begin{minipage}{%
  (\linewidth-2\fboxrule-2\fboxsep)}}%
  {\end{minipage}\end{Sbox}\fbox{\TheSbox}\end{table}}
```

のように `ftable` や `ffigure` のような環境が定義できます。この出力例が表 8.5 となります。

`fancybox` では `center` 環境のように行揃えを行う環境があります。これを枠で囲むには `Bcenter`, `Bflushleft`, `Bflushright` 環境を使います。

表 8.5 L^AT_EX の歴史L^AT_EX 2.09 L^AT_EX 2_ε L^AT_EX 3

```
\fbox{\begin{Bcenter}
ここは \\ 中央揃えで \\
  枠付きです.
\end{Bcenter}}
```

```
ここは
中央揃えで
枠付きです.
```

```
\fbox{\begin{Bflushleft}
ここは \\ 左揃えで \\
  枠付きです.
\end{Bflushleft}}
```

```
ここは
左揃えで
枠付きです.
```

```
\fbox{\begin{Bflushright}
ここは \\ 右揃えで \\
  枠付きになります.
\end{Bflushright}}
```

```
ここは
右揃えで
枠付きになります.
```

minipage 環境の中で行揃えの命令を使う事も考えられますが、その場合は横幅を指定しないといけないので、これらの環境は便利でしょう。箇条書き環境も同様に minipage の中に入れて \fbox で囲む事も考えられますが、fancybox では Bitemize, Benumerate, Bdescription の三つがすでに定義されているので、これらの環境を枠で囲みます。

```
\fbox{\begin{Benumerate}
\item 北海花子.
\item 太郎君と花子さん.
\item 日本太郎.
\end{Benumerate}}
```

```
1. 北海花子.
2. 太郎君と花子さん.
3. 日本太郎.
```

\verb 命令を枠で囲むときは \VerbBox 命令を使います。

```
\VerbBox{\fbox}{
\verb|\VerbBox{\fbox}{\verb+{+}}}
```

```
\VerbBox{\fbox}{\verb+{+}}
```

verbatim 環境を枠で囲む場合はやはり先に Sbox 環境で囲んで \TheSbox を枠で囲むようにします。この場合 fverbatim 環境を定義したほうが便利でしょう。ただし、

この場合 `\VerbatimEnvironment` 命令と `Verbatim` 環境を併せて使わないとエラーになります。

```
\newenvironment{fverbatim}[1]%
  {\VerbatimEnvironment
  \begin{Sbox} \begin{minipage}{#1}%
  \begin{Verbatim}}{\end{Verbatim}}%
  \end{minipage}\end{Sbox}}
  \fbox{\TheSbox}}
\begin{fverbatim}{.8\linewidth}
このべた書き環境は枠で囲まれて
出力されます。
\end{fverbatim}
```

このべた書き環境は枠で囲まれて
出力されます。

枠付の箱を出力するために \LaTeX 標準の `framebox` 環境や `\fbox` などが使用できますが、もうすこし、角がまるかったり、数式などにも使えると便利です。ある程度は好き好きシリーズの初級編で取り扱っているので、そちらを参照して下さい。ここでは Timothy Zandt 氏による `fancybox` をもう少し細かく見てみましょう。

```
\boxput*(x,y){<下になる要素>}{<上になる要素>}
```

という `\boxput` 命令が使えます。使用例を参考にして下さい。 `\boxput` に星を付けると重ね合わせる順番が逆になります。

```
\usepackage{fancybox,color}
\boxput(0,0){\color{blue}\Huge Draft}
{\color{red}\parbox{\linewidth}{%
こちらはなんでもかんでも\LaTeX
でがんばります。 \LaTeX は最高で
す。とにかく一度使ってみて下さい。}}
```

こちらはなんでもかんでも \LaTeX でがんばり
ます。 \LaTeX は最高です。とにかく一度使っ
て下さい。

Draft

他にもページごと枠で囲む `\fancyput` や `\thisfancyput` などもあります。

```
\fancyput*(x,y){<文章要素>}
\thisfancyput*(x,y){<文章要素>}
```

ここでは触れませんが `verbatim` 環境系の拡張がされています。

このほかにも `fancybox` にはファイルから読み込んだ行をべた書き環境に出力する命令なども用意されています。

8.10 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**空き**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、本書では二つを区別して表します。

▼ 8.10.1 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 8.6 の通りです。表 8.6 は基本的に空きの前後での改行を行っても良い事になっています。

表 8.6 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>\qqquad</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`{\qqquad}`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 8.6 の命令は改行を許しますが表 8.7 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。

Donald~E. Knuth made \TeX\@.
Leslie~Lamport made \LaTeX\@.

Donald E. Knuth made T_EX. Leslie Lamport made L^AT_EX.

表 8.7 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>	-3/18 quad 分の空き
<code>~</code>	適切な単語間空白

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]}
```

アスタリスクを付けると行頭・行末でも使えるようになります。

```
日本太郎\hspace{2zw}ならば\par          日本太郎   ならば
\hspace*{-2zw}である。                  である。
```

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用さ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。

▼ 8.10.2 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`, `\midskip`, `\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。「見映えが損なわれない程度におおよそ指定した長さの空気を挿入してほしい」というような意味合いを持っています。垂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所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気を挿入するのは好ましくないとされます。

長さを自分で指定して空気を挿入する場合は `\vspace*` と `\addvspace` が使えます。

表 8.8 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

```
\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
挿入されると思われます。

8.11 伸縮する糊

L^AT_EX ではバネのように伸縮する糊のような便利な道具があります。これは活版印刷時代における活字職人が経験で挿入する**込め物**に似ています。

L^AT_EX においてもある領域の中での空きを制御するためとか、適切な表示をするための込め物が使われます。L^AT_EX で使われる込め物は**グルー**と呼ばれています。グルーは伸縮自在でバネのように要素と要素をくっ付けるので伸縮する糊とも呼ばれています。

グルーは特別な単位 `'fil'` と `'fill'` によって定義されています。`'fil'` は良い^{あんばい}按排で 0 からかなり大きい空きまで挿入する働きをします。`'fill'` は `'fil'` よりも大きく無限に近い空気を挿入する働きをします。`'fil'` は `'fill'` よりも弱いので `'fill'` のほうが優先されます。

この空き `{\hfil}` は `{\hfill}` にかき消され
ません。 `\par`
この空きは `{\hfil}` 恐らく `{\hfill}` にかき消
されます。

この空き は にかき消されません。
この空きは恐らく にかき消されます。

`'fil'` と `'fill'` の二つのグルーの両方を使っている場合は空気が挿入されない場合

表 8.9 L^AT_EX で使用できるグルー

命令	意味
<code>\hfil</code>	水平方向にかなり大きく伸びる空き
<code>\hfill</code>	水平方向に無限に大きく伸びる空き
<code>\hss</code>	水平方向にかなり大きく伸び縮みする空き
<code>\vfil</code>	垂直方向にかなり大きい空き
<code>\vfill</code>	垂直方向に無限に大きい空き
<code>\vss</code>	垂直方向にかなり大きく伸び縮みする空き

がありますから気を付けてください。

もう少し相対的なグルーを挿入するには`\stretch`を使います。

```
\stretch{(整数)}
```

これは‘fill’の何倍かのグルーを挿入しても良いようにするために使用できます。

```
hoge\hspace{\stretch{3}}hoge%
\hspace{\stretch{1}}hoge.\par      hoge          hoge          hoge.
hoge\hspace{\fill}hoge\hspace{\fill}■ hoge          hoge          hoge.
hoge.\par
```

`\fill`は`\hspace`や`\vspace`の引数に使う事ができるグルーです。例えばページの最上部に無限に近いグルーを挿入するときは以下のようにします。

`\vspace*{\fill}`を使ってもこの出力例では伸びません。 `\vspace*`を使ってもこの出力例では伸びません。

▼ 8.11.1 空白ではないグルー

`\hfil`や`\hfill`は空白を挿入するグルーでした。そうではなく、与えられた領域を程よく文字列や要素で埋めてくれる込め物もあると便利です。目次でも使われているのがお分かりになるでしょう。これらを**リーダ**と呼びます。L^AT_EX であらかじめ定義されているリーダは表 8.10 となります。

‘fill’ですから無限に伸びます。`\dotfill`と`\hrulefill`は`lTplain.dtx`で、

表 8.10 主なリーダー

命令	出力例
<code>\dotfill</code>
<code>\hrulefill</code>	—————
<code>\leftarrowfill</code>	←—————
<code>\rightarrowfill</code>	—————→
<code>\downbracefill</code>	————— }—————
<code>\upbracefill</code>	————— {—————

それ以外は `fontdef.dtx` にて定義されています。段落中で使うと段の終わりまで伸びます。

```
\makebox[5zw][l]{未\dotfill 来}\      未.....来
\hrulefill\                            —————
北海\leftarrowfill 道\                北海 ←————— 道と
北方領土\rightarrowfill\par          北方領土 —————→
```

これらのリーダーは `\leaders` という命令によって定義されています。自分でこのような命令を定義したいときは `\leaders`, `\cleaders`, `\xleaders` の三つが使えます。例えば領域を「未来」で埋めるような命令ならば次のようになります。

```
\newcommand{\hogefill}{\leavevmode
\leaders\hbox{未来}\hfill\kern0pt}  未来未来未来未来未来未来未来未来未来
\hogefill\par                        そう 未来未来未来未来未来未来未来未来 だよ。
そう\hogefill だよ。 \par
```

8.12 付録の追加

文書の最後に付録としてプログラムリストを載せるとか、本文とは直接的に関係のない資料を載せるときは `\appendix` 命令を使うか、`appendix` 環境を使うかの 2 通りの方法があります。`appendix` 環境を使う場合は付録の範囲を指定できます。

```
\begin{appendix}
<追加する内容>
\end{appendix}
```

この命令を付けた後の文章は付録として扱われ、見出しの番号付けが自動的に大文字のアルファベットに変更され、‘A’からカウントされるようになります。あとは通常通り見出しの定義をして文章を記述するだけです。

8.13 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。大規模な文書の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) に分ける事が考えられます。`\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事もできます。

```
#!/platex
\documentclass[dvipdfmx,papersize]{jsbook}% クラスファイル
\usepackage{amsmath,amssymb,bm,verbatim,listings}% 必要なマクロ
\includeonly{2joron}% ある章だけを読み込む
\begin{document}
\maketitle
\tableofcontents
\listoffigures
\listoftables
\frontmatter% 前付
\include{0preface}% 前書き
\include{1thanx}% 謝辞
\mainmatter% 本文
%\doublespacing %
\include{2joron}% 序論
\include{3honron}% 本論
\include{4keturon}% 結論
%\singlespacing %
\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}
```

8.14 翻訳作業

しばしば日本語ではない言語で書かれた文書を訳す作業があります。運良く原書の L^AT_EX の原稿が手に入ったとすると、作業は幾分楽になります。例えば以下のような原稿があったとします。

```
Hello, everyone! I'm a student at Future University Hakodate.
Today, please let me talk about my future plan.
First, ...
```

これを工夫せずに普通に翻訳すると次のようになるでしょう。

```
皆さん、こんにちは。私は公立はこだて未来大学の生徒です。
今日は私の未来計画についてお話したいと思います。まず、
```

どうせなら原書の英文も削除したくありません。

```
%Hello, everyone! I'm a student at Future University Hakodate.
皆さん、こんにちは。私は公立はこだて未来大学の生徒です。
%Today, please let me talk about my future plan.
今日は私の未来計画についてお話したいと思います。
%First, ...
まず、
```

このように入力すると英文と和文の対応が取れて分かりやすいでしょう。ワープロソフトではマネのできない芸当です。1 行ずつに分ける必要はなく、非常に長い文章の場合は 1 段落ごとに対応させるのも良いでしょう。

8.15 用語の統一

大規模な文書の場合、**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次のようにします。

```
\newcommand*{\FUN}{公立はこだて未来大学}
```

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事になります。

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*{(コマンド名)}{(文字列)\xspace}
```

```
\usepackage{xspace}
```

```
\newcommand*\LC{Logical OR}% (×) Logical OR is good? ‘Logical OR’ is also
\LC is good? ‘\LC’ is also ok. ok. Logical OR is funky! ‘Logical OR’ is
\renewcommand*\LC{Logical OR\xspace} also ok.
\LC is funky! ‘\LC’ is also ok.
```

8.16 色の指定——color

\TeX / \LaTeX では白と黒の 2 色しか理解できません。他の色を表現しようと思えば `color` パッケージなどの力を借りてデバイスドライバに全てを任せる事になります。ですから色はドライバに依存しますし、プリンタがモノクロプリンタならばどうがんばってもグレースケールのページしか出力できません。

色には**色相・明度・彩度**の三つの要素があります。色相とは青や緑などの波長の種類、明度はその波長の明るさ、彩度は黒色の少なさを表します。

▼ 8.16.1 要素に色を付ける

\LaTeX は白と黒の 2 色しか理解できません。ですからそのほかの色を付ける場合はデバイスドライバに全てを任せます。グレースケールでも同様です。色を付けるために

は `graphics` パッケージと同封されている David Carlisle 氏が作成した `color` パッケージを使います。 `color` を使うにはまず `graphicx` と同じように使用するデバイスドライバを指定します。 `Dvipdfmx` などを使っているときは次のようにします。

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

パッケージオプションとして次のようなものがあります。

usenames `color` パッケージで定義されている色の名前を全て使えるようにします。

dvipsnames `dvips` で使用できる色を名前を指定して使えるようにする。

▶ **問題 8.6** 以下のファイル `colortest.tex` をタイプセットし、さらに `Dvipdfmx` により PDF に変換し、その出力結果を吟味してください。

```
#!/F=hoge; platex $F.tex && dvipdfmx $F && open $F.pdf
\documentclass[10pt,dvipdfmx,papersize]{jsarticle}
\usepackage{typelec,multicol}
\usepackage{color}
\pagestyle{empty}
\begin{document}
\renewcommand \DefineNamedColor[4]{%
  \textcolor[#3]{#4}{\rule{2em}{1em}}\space{#2}\}
\parindent=0pt
\begin{multicols}{3}
  \input{dvipsnam.def}
\end{multicols}
\end{document}
```

実際にご自分が持っている（カラー/モノクロ）のプリンタで上記のファイルを印刷してください。さらに印刷したものとディスプレイでプレビューしているファイル `colortest.pdf` を見比べてください。カラーで出力した方は一度モノクロでも印刷してみてください。

もしも、差があるようであれば、どのような差があるのか頭の中で整理してください。

色の指定には四つあります。

rgb Red, Green, Blue の 3 色を混ぜ合わせて加法混色で色を指定します。それぞれの色は 0 から 1 のあいだで指定します。

cmymk Cyan, Magenta, Yellow, Black の 4 色を混ぜ合わせて減法混色で色を指定します。これも 0 から 1 のあいだで指定します。

gray グレースケールで 0 から 1 のあいだで濃さを決めます。

named あらかじめ定義された名前を使う事を意味します。

新規に色の名前を定義するときは `\definecolor` を使います。

```
\definecolor{<名前>}{<種類>}{<値>}
```

〈種類〉には上記の四つが選択できます。例えば以下のように定義します。

```
\definecolor{MyGray}{gray}{0.85}
\definecolor{MyRed}{rgb}{0.3,0,0}
\definecolor{MyYellow}{cmyk}{}
```

文字列の色は `\textcolor` を使って色を指定します。

```
\textcolor{<色>}
\textcolor[<種類>]{<値>}{<文字列>}
```

ある範囲中の要素には `\color` 命令で指定します。

```
\color{<色>}
\color[<種類>]{<値>}
```

```
\textcolor{Gray}{この文字は灰色}\
\textcolor[rgb]{1,0,0}{赤らしく}\
{\color{Gray}{\fbox{枠も文字も灰色
です。}}}
```

この文字は灰色

赤らしく

枠も文字も灰色です。

ページの背景色を指定するときは `\pagecolor` 命令を使います。

```
\pagecolor{<色>}
\pagecolor[<種類>]{<値>}
```

これは命令を使ったそのページからずっと色の指定が有効ですので、1 ページだけの色を変更したいときはページの切り替わるだろう部分で色を元々の色に戻します。

枠付きの箱の背景を指定できる `\colorbox` 命令があります。

```
\colorbox{<色>}{<文字列>}
\colorbox[<種類>]{<値>}{<文字列>}
```

```
\colorbox{red}{背景が赤になる}\
\colorbox[gray]{.8}{背景は灰色}\
\colorbox[black]{\color{white}}%
背景が黒で文字は白}
```

背景が赤になる

背景は灰色

背景が黒で文字は白

さらに枠の色とその背景の色を指定できる `\fcolorbox` もあります。

```
\fcolorbox{<枠の色>}{<背景色>}{<文字列>}
\fcolorbox[<種類>]{<枠の色の値>}{<背景色の値>}{<文字列>}
```

`\fcolorbox` では `\fbox` と同じように `\fboxrule` と `\fboxsep` を調整できます。

```
\fcolorbox{red}{blue}
{枠が赤で背景が青}\
```

枠が赤で背景が青

```
\fcolorbox[gray]{.5}{.8}
{枠も背景も灰色}\
```

枠も背景も灰色

```
{\fboxrule=2.4pt\fcolorbox{blue}
{red}{枠の線幅の調節}}
```

枠の線幅の調節

8.17 プログラムソースの挿入——listings

プログラムなどのソースコードを載せるときに、`verbatim` だけでは寂しいと感じる方も多いと思います。変数やコメントなどをうまく整形するプログラムやマクロパッケージがあります。アルゴリズムの行数を文中で指定したいときには行数も表示されとうれしいものです。Texinfo のように変数名や関数名なども半自動的に索引に追加されると便利です。ソースコードの整形プログラムの中で私が良いと思うのが Carsten Heinz 氏が作成した `listings` です。これは正式には日本語が通りませんが、吉永徹美氏が作成した自称強引なマクロで切り抜ける事ができます。これに関しては奥村晴彦氏の掲示板の『汎用的な浮動体』^{*5} という一連の書き込みを参照してください。結果としてこれをまとめたファイルを置いておきます^{*6}。

他にも `lgrind`, `morevrb`, `progc` などの類似品がありますがそれらの包括的なパッケージが `listings` です。

`listings` を使用するときにはプリアンブルに次のようにします。

```
\usepackage[オプション]{listings}
```

このときに指定するパッケージオプションは次のようなものがあります。

draft ソースコードを出力しないようにします。ドキュメントクラスオプションですでに `draft` を指定している場合はそれが反映されるので、さらにオプションを記述する必要はありません。

^{*5} <http://oku.edu.mie-u.ac.jp/~okumura/texfaq/qa/21172.html>

^{*6} <http://tex.dante.jp/jou1/jlisting.zip>

final draft オプションを無効にし、実際にソースコードを整形します。
savemem \TeX/\LaTeX のメモリをなるべく消費しないようにします。

listings は \TeX のメモリをたくさん使いますので原稿執筆の段階では *draft* オプションを付けたほうが良いでしょう。

listings の使い方は大きく分けて二つです。一つは `lstlisting` 環境にソースコードを記述する方法、二つ目はプログラムのソースコードをファイルから入力する方法です。一つ目の \LaTeX の原稿に直接記述する方法は次のような文法になります。

```
begin{lstlisting}[(設定 1, 設定 2,...)]
ソースコード
end{lstlisting}
```

例として C のソースコードであれば次のようにします。

```
\begin{lstlisting}[language=C]
int main (void){
    printf ("Hello, World!!\n");
}
\end{lstlisting}
```

二つ目の方法としては `\lstinputlisting` 命令を使ってファイルからソースコードを読み込みます。

```
\lstinputlisting[(設定 1, 設定 2,...)]{<ファイル名>}
```

〈設定〉を毎回指定していたのでは疲れますので `\lstset` 命令で次のように設定しておけば全てのソースコードに共通の設定をすることができます。

```
\usepackage[dvips]{color}
\lstset{%listings の表示設定
    frame=tbrl,%枠を上下左右に表示する
    backgroundcolor={\color[gray]{0.85}},%背景を灰色に
    numbers=left,%行番号を左に
    numberstyle=\scriptsize,%
    stepnumber=1,%1行おきに行番号を
    numbersep=1zw}%ソースと行番号の間隔
```

この設定の場合は上下左右に枠を表示させ、背景色を薄いグレーにし、行番号を左側の 1 行おきに表示するようにします。

いくつかの設定をひとまとめにしたいときは `\lstdefinestyle` 命令を使ってスタイルを定義します。

```
\lstdefinestyle{<スタイル名>}{<設定>}
```

例えば行番号に関しては左側に 1 行おきに表示させたいので

```
\lstdefinestyle{number1}{numbers=left, numberstyle=\scriptsize,%
  stepnumber=1, numbersep=1zw}
```

という定義をして

```
\begin{lstlisting}[language=C,style=number1]
```

のように指定したり、または特定の言語に共通のスタイルを用いるときは

```
\lstdefinestyle{C}{language=C,style=number1}
```

として

```
\begin{lstlisting}[style=C]
```

と使う事もできます。

ある言語のソースコード用に新しい環境を定義できます。

```
\lstnewenvironment{<環境名>}{<始めの設定>}{<終わりの設定>}
```

例えば C 言語用の環境を新たに定義するときは次のようにしても良いでしょう。

```
\lstnewenvironment{C}{%
  \lstset{language=C,style=number1}}{}
```

▷ **例題 8.7** ソースコードを浮動体 (float) として出力する事もできます。実際にタイプセットし、その結果を吟味してください。

```
\begin{lstlisting}[language=C,float,caption={listings の使用例}]
int main (void){
  printf ("Hello, World!!\n");
}
\end{lstlisting}
```

▼ 8.17.1 言語の設定

listings では非常に多くの言語用の設定が定義されています。あらかじめ定義されていない言語は自分で新たに定義する事もできます。あらかじめ定義されている言語は表 8.11 の通りです。

表 8.11 listings で使用できる主な言語

Assembler	Basic	C
C++	Cobol	csh
[Sharp]C	Delphi	Fortran
HTML	IDL	Java
ksh	[LaTeX]TeX	Lisp
Logo	make	Mathematica
Matlab	MetaPost	MuPAD
Octave	Pascal	[plain]TeX
Perl	PHP	Prolog
Python	Ruby	Scilab
SQL	tcl	[tk]tcl
TeX	VBScript	[Visual]Basic
[Visual]C++	VRML	XML

角括弧を含む言語の指定は波括弧の中に入れて指定するのが安全です。

```
\begin{lstlisting}[\language={ [LaTeX]TeX}]
```

言語があらかじめ定義されていない場合、`\lstdefinlanguage` 命令で自分で言語の定義をすることができます。

```
\lstdefinlanguage{<言語名>}{<設定>}
```

▷ **例題 8.8** 次のような定義をした言語 ‘CCC’ における文法について考えてみてください。

```
\lstdefinlanguage{CCC}{
  morekeywords={short, int, long, float, double},%予約語
  sensitive=false,
  morecomment=[1]{//},%行末コメント
  morecomment=[s]{/*}{*/},%範囲コメント
  morestring=[b]" ,%文字列
  morestring=[d]',%文字}
```

▶ **問題 8.9** 標準の listings の字詰めや行送りはローマン体でなるべく等幅になるように字詰めをしています。そちらのほうが正しい設定なのでしょうが、見栄えを考えると次のようにしたほうが良いかもしれません。

```
columns=[1]{fullflexible}
```

タイプライタ体で出力しない場合は等幅になりませんので、ソースコードとしては不適切な設定かもしれません。ご自分で `fullflexible` 以外にも `fixed`, `flexible` などを ‘`columns`’ に指定して判断して下さい。

▼ 8.17.2 主な設定値

空白などに関する設定です。

`lineskip`=〈長さ〉 行間にさらに加える空きを調節します。標準は 0pt。

`firstline`=〈整数〉 入力されているソースコードについて、どの行から読み込むかを設定します。標準は 1。

`lastline`=〈整数〉 ソースコードのどの行まで読み込むかを設定します。標準は 9999999。

`tabsize`=〈整数〉 ソースコード中のタブ文字を何文字分にするかを設定します。標準は 8 です。

`showtabs`=〈`true`|`false`〉 タブ文字を可視・不可視にします。

`tab`=〈文字列〉 タブ文字を〈文字列〉に置き換えて表示します。\$`\Longrightarrow`\$ (→) などを使う方法もあります。

`showspace`=〈`true`|`false`〉 スペースを可視 ()・不可視 () にします。

`linewidth`=〈長さ〉 ソースコードの文章幅を指定します。標準は `\linewidth` です。

`xleftmargin`=〈長さ〉 左側の余白を指定します。標準は 0pt です。

`xrightmargin`=〈長さ〉 右側の余白を指定します。標準は 0pt です。

`breaklines`=〈`true`|`false`〉 ソースコードの文章幅よりも長い文字列を自動的に折り返すかを指定します。標準は `false` です。

`prebreak`=〈文字列〉 行頭に挿入する文字列を指定します。

`postbreak`=〈文字列〉 行末の挿入する文字列を指定します。例えば行末の改行の位置を特別に示したいときは ‘`postbreak=\return`’ のようにすると良いでしょう。

言語の設定です。

`language`={〈言語〉} あらかじめ定義されている主な言語については表 8.11 をご覧ください。

文字の表示の仕方の設定です。

`basicstyle=〈スタイル〉` 通常のソースのスタイルを設定します。〈スタイル〉には `\small` や `\ttfamily` などのフォントの宣言をする命令が使えます。

`commentstyle=〈スタイル〉` コメントのスタイルを設定します。

`stringstyle=〈スタイル〉` ソースコード中の文字列のスタイルを設定します。

`keywordstyle=〈スタイル〉` キーワード、プログラミング言語で言えば予約語のスタイルの設定です。

行番号の表示の設定です。

`numbers=(none|left|right)` 行番号をどのように表示するかの設定です。

`stepnumber=〈数字〉` 何行おきに行番号を表示するかを設定します。

`numberstyle=〈スタイル〉` 行番号のスタイルを指定します。

`firstnumber=(auto|last|〈数字〉)` 行番号の開始の数字を指定します。

浮動体などに関する設定です。

`float=(htbp の部分集合)` ソースコードを浮動体として出力します。

`caption=(文字列)` 見出しの文字列を指定します。

`label=(文字列)` `\ref` 命令で参照できるラベルを作成します。

`captionpos=(t|b)` 見出しの位置を指定します。

以下の命令はソースコードを目次として出力するときなどの命令です。適宜再定義してください。

`\lstlistoflistings` ソースコード目次を出力します。これは `caption` を付けたソースコードが出力されます。

`\lstlistlistingname` ソースコード目次の見出しです。標準は ‘Listings’ です。

`\lstlistingname` 見出しの前の文字列を指定します。標準は ‘listing’ です。

`\lstlistlistingname` 命令と `\lstlistingname` は次のように定義しておくくと日本語の文書でも違和感がないと思います。

```
\renewcommand{\lstlistlistingname}{ソースコード目次}
\renewcommand{\lstlistingname}{ソースコード}
```

罫線枠に関する設定です。

`frame`= \langle none|single \rangle ソースコードの周りに表示する罫線枠の設定です。他にも `shadowbox`, `leftline`, `topline`, `bottomline`, `lines` が指定できます。

`frame`= \langle trblTRBL の部分集合 \rangle 上下左右の罫線の表示を `t`, `b`, `l`, `r` で指定します。

`framerule`= \langle 太さ \rangle 枠の線の太さを指定します。標準では 0.4pt です。

`framesep`= \langle 長さ \rangle 枠と他の要素との空きを指定します。

▶ **問題 8.10** 枠を付けて表示する例としては次のようになります。

```
\begin{lstlisting}[framerule=.4pt,frame=shadowbox,language=C]
void main (void){
    int a = 3; int b = 5;
    printf ("a + b = %d.\n", a + b);
}
\end{lstlisting}
```

出力結果がソースコード 8.1 のようになります。

ソースコード 8.1 枠の調整

```
void main (void){
    int a = 3; int b = 5;
    printf ("a $\square$ + $\square$ b $\square$ = $\square$ %d.\n", a + b);
}
```

色に関する設定です。

`backgroundcolor`= \langle 色 \rangle `color` パッケージにおける宣言型の命令が使えます。

`rulecolor`= \langle 色 \rangle 罫線枠の色を指定します。

`rulesepcolor`= \langle 色 \rangle 枠線の色と他の要素のあいだの色を指定します。

色を着ける例としては次のようになります。

```
[frame=single,backgroundcolor={\color[gray]{.85}},
rulecolor={\color[cmymk]{0,1,1,.3}},language={C}]
```

結果的にはソースコード 8.2 となります。

ソースコード 8.2 色の調整

```
void main (void){
    int a = 3; int b = 5;
```



```
printf ("a_+_b_=%_d.\n", a + b);
}
```

ソースコードは浮動体として宣言できます。

```
\lstinputlisting[language={Prolog},caption=Prolog のソースコードを%
\textsf{Listings}]を使って挿入する例,label=src:prolog]{filename.pl}
```

このようにすれば相互参照も可能になります (ソースコード 8.3)。

ソースコード 8.3 Prolog のソースコードを Listings を使って挿入する例

```
% File: prolog.pl
% Author: Toru Watanabe
% Date: Saturday, Aug. 9th, 2003
% Original Source Information:
% File: Micro Expert System.
% Ian Frank, July 2001, updated May 2002
% (Based on original by Alison Causey)
% Dynamic userfact/1
:- dynamic(userfact/1).
% Clear userfact/1
clear_facts :-
    retract(userfact(_)),
    clear_facts.
clear_facts :-
    write('OK, removed userfacts').
% New Operators
:- op(975,fx, if).
:- op(950,xfy, then).
:- op(925,xfy, and).
%% If All people come to X's party, then X is happy.
rule(if all_come_the_party(X) then happy(X) ).
qtext(good(X,wed_class) , ['!s_',X,'s_!Wednesday_class_good?'] ).
%% If X is happy, this problem would be succeed.
atext(happy(X) , ['Everyone_comes_to_the_party_and_',X,'_!s_!happy!!'
]).
%
prove(Goal) :-
    bchain(Goal),!, % bchain to check if true.
    atext(Goal, Text),% get hold of appropriate text.
    write_list(Text). % write out the recommendation
%
prove(_ ) :-
```

```

write_list(['The goal does not seem to be true.']).
%
yesno(Text) :-
  write_list(Text),
  write_list(['(y/n)']),
  get(X),
  X =:= 121.

```

8.18 URL の記述——url

最近では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 L^AT_EX でやろうと思えば `\verb` 命令が使えると思うのですが脚注の中で使えないとか、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\folder{\begingroup
  \urlstyle{tt}\Url}
\url{ftp://www.any.com/dir/file.htm}
\email{name@server.ac.jp}という
メールアドレスがあったので
\folder{/usr/local/bin/}
のファイルを消した。

```

ftp://www.any.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/ のファイルを消した。

8.19 ハイパーリンクの実現——hyperref

L^AT_EX でも HTML のように相互参照にハイパーリンクを実現できます。Sebastian Raetz 氏が作成した `hyperref` [96] を使う事になると思います。これは `\special` 命令

によってハイパーリンクを実現する HyperTeX を使いやすくしたものと考えて良いです。

ETEX と HTML を足して 2 で割ったような文法で、次のようなマクロが提供されています。〈URL〉の書き方は HTML とまったく同じで、拡張子や#なども忘れないでください。

```
\hypertarget{〈name〉}{〈文字列〉}
```

\hypertarget は〈name〉を名札として〈文字列〉にラベルを付けます。ラベルをつけたものは \href などにより参照できます。

```
\href{〈URL〉}{〈文字列〉}
```

\href はファイルやラベルなどの〈URL〉を参照します。〈URL〉には次のように絶対パスでファイルやサイトを指定することができますし、相対パスで指定する事もできます。

```
\href{http://www.server.co.jp/~user/index.htm}{ここをクリック}
```

URL にチルダや特殊な記号があるからといって、特別な事はしなくても良いようです。アドレス中にピリオドが二つあると拡張子を誤認するかもしれないので、一番最後のピリオド以外をアンダーバーに変えるとどうにかなるかもしれません。

```
\hyperimage{〈URL〉}
```

\hyperimage は画像ファイルの〈URL〉を指定し、該当するファイルがあればそれを張り込みます。

8.18 節で紹介した url と同等の \url 命令も使えます。hyperref の \url を使うと自動的にハイパーリンクが作成されます。

▼ 8.19.1 パッケージオプション

hyperref はほとんど全てのパッケージオプションを keyval を使って ‘〈項目〉=〈値〉’ のような指定で設定します。ブール値ならば〈値〉を省略できます。\usepackage 命令の任意引数に対して〈オプション〉を渡せばそれが hyperref のパッケージオプションになります。

```
\usepackage[pdfpagemode=FullScreen,colorlinks=true,dvipdfm]{hyperref}
```

このようにすると PDF 文書を開いたときにフルスクリーンで表示し、リンクのある文字の色が変更され、DVI ドライバには dvipdfm を使うという設定になります。hyperref

で使用できる主なパッケージオプションを紹介しておきます。特に断りがない限り以下のオプションはブール値です。

```
breaklinks=false
breaklinks=true
```

上記のように有効 ('true') か無効 ('false') で指定できるようになっています。ただ単にそのオプションを指定するとそのオプションが有効になります。

デバイスドライバ デバイスドライバとしては *dvips*, *ps2pdf*, *latex2html*, *tex4ht*, *pdftex*, *dvipdfm* などが使えます。

用紙サイズ 用紙サイズを *a4paper*, *a5paper*, *b5paper* などで指定してデバイスドライバに応じた設定をします。

ハイパーリンクに関する設定です。

breaklinks ハイパーリンクが複数行にまたがっても良いようにします。使用したほうが良いでしょう。

colorlinks ハイパーリンクがある事を枠ではなく、文字列に色を付けて示します。可読性を考えて使用すると良いでしょう。

PDF しおりに関する設定です。

bookmarks PDF しおりを作成するようにします。

bookmarksopen PDF しおりを展開するかどうかの設定です。dvipdfm などはパッケージ側で指定してもプログラム側で対応していないかもしれません。角藤亮氏の配布している dvipdfm と Dvipdfmx にはコマンドラインオプション-E によって展開できます。

bookmarkopenlevel=〈数値〉 PDF しおりを展開する深さを指定します。これは L^AT_EX における見出しと同じ〈数値〉を使いますから表 3.3 を参照してください。

bookmarksnumbered PDF しおりにおいて見出し番号を含めます。

bookmarkstype=〈種類〉 (種類)には PDF しおりに使うべき目次情報を指定します。PDF しおりのために特別なしおり情報をすでに別ファイル(*file*).foo に作成しているならば〈種類〉には 'foo' を指定します。

PDF しおりに特殊なコマンドが含まれているときはうまくしおりが作成されませんので `\texorpdfstring` 命令を使います。

```
\texorpdfstring{ $\langle LATEX \text{での表記} \rangle$ }{ $\langle PDF \text{での表記} \rangle$ }
```

これを使うのは例えば見出し用のコマンドに数式などが含まれるときです。以下のようになるとうまく処理できないでしょう。

```
\section{この\hoge について}
```

そのため、`\texorpdfstring` を使うのが良い事になります。

```
\section{この\texorpdfstring{\hoge}{hoge}について}
```

デバイスドライバとしては `dvipdfm` ではなく `Dvipdfmx` を使うと良いでしょう。その場合はオプションとして `'dvipdfmx'` ではなく `'dvipdfm'` を渡します。

```
\usepackage[dvipdfm,bookmarks=true,%
bookmarksnumbered=true,bookmarkstyle=toc]{hyperref}
```

上記のようにすると `\file.toc` から PDF しおりが見出し番号付きで PDF に追加できます。この場合原稿の先頭で PDF に対して渡すべき文字コードを指定します。

```
\ifnum 42146=\euc"A4A2
\AtBeginDvi{\special{pdf:tounicode EUC-UCS2}}\else%"
\AtBeginDvi{\special{pdf:tounicode 90ms-RKSJ-UCS2}}\fi
```

私は大抵、以下のように設定しています。

```
\ifnum 42146=\euc"A4A2
\AtBeginDvi{\special{pdf:tounicode EUC-UCS2}}\else%"
\AtBeginDvi{\special{pdf:tounicode 90ms-RKSJ-UCS2}}\fi
\usepackage[dvipdfm,bookmarks=true,%
bookmarkstyle=toc,bookmarksnumbered=false,%
bookmarksopen=true,colorlinks=true,%
linkcolor=blue,citecolor=blue,filecolor=blue,%
menucolor=magenta,pagecolor=blue,urlcolor=blue,%
backref=page]{hyperref}
```

PDF ファイルに対して「文書情報」というものを追加したければ `hyperref` を読み込んでから次のような設定をします。

```
\special{pdf:docinfo <<
/Author ( 著者を書く )
/Title ( 主題を書く )
/Subject ( 副題を書く )
/Creator ( どのプログラムを使ってPDFを作成したのか )
/Keywords ( キーワード, 複数個, 指定, 可能)
>>}
```

丸括弧の中には適当な情報を追加します。これを `Dvipdfmx` で PDF に変換し `pdfinfo` でその PDF 文書情報を見る事もできます。パッケージオプションに次のようにしても同じ事になります。

```
\usepackage[dvipdfm,pdftitle={主題},%  
pdfsubject={副題},pdfauthor={著者},%  
pdfkeywords={キーワード}]{hyperref}
```

8.20 原稿の執筆支援

L^AT_EX 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。

▼ 8.20.1 入力支援統合環境——YaT_EX

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 YaT_EX が便利です。詳しい事は YaT_EX のウェブページ^{*7} から情報を集めてください。

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。

▼ 8.20.2 原稿のコンパイル支援——Make

L^AT_EX はコンパイル型の言語です。

ソースファイルを L^AT_EX プログラムに渡せば人間が読める媒体に変換します。この L^AT_EX のソースファイルに対して Make と呼ばれるプログラムを使う事で原稿の再コンパイルを支援してくれます。これは一つのグループになったソースファイルの依存関係をはっきりとさせて、再コンパイルを自動化するプログラムです。

Make は Unix 系 OS 主流のプログラムですから、Unix 系 OS でなければ実行させるのが難しいかもしれませんが、あなたの OS が GNU Linux ならば簡単にインストールできるはずです。Windows の方は Cygwin と呼ばれる Windows 上で動作させることができる擬似 Unix 環境がありますのでこれを導入してみてください。

コンソールなどから次のように

```
$ make
```

とすれば次のような表示が出るでしょう。

^{*7} <http://www.yatex.org/>

```
「make: *** No targets specified and no makefile found. Stop.」
```

ゲームを始める準備は整っているようです。表示されないならば、近くの詳しい人に聞いてみましょう。

まずは簡単な例を示しますので、Make の基本を見てみましょう。

```
all: file.dvi
file.dvi: file.tex
—————→latex file
```

これは以下のような文法に則り記述されています。

```
all: <結果的に作りたいもの>
<作りたいもの>: <それに依存するもの>
—————→処理内容
```

上記のソースにおいて ‘—————→’ はタブ文字をあらわします。これは非常に重要な点です。Make での変数の定義は単純に次の書き方になります。

```
<変数>=<値>
```

同じ記述が何度も繰り返される場合はそれを変数として定義できます。

```
MENDEX=mendex -d jisyo.dic -g
TEX=latex
DVIPS=dvipsk -Pdl -t a4
DVIPDFM=dvipdfmx -f hiarginox.map -p a4 -V 4 -z 5
```

LaTeX では主となる原稿に対して章ごとに分割されたファイルを `\include` 命令で読み込んでいる事があるでしょう。この場合 LaTeX でタイプセットした後に生成される DVI ファイル (`file`).dvi は主となる原稿 (`file`).tex と分割されたファイルに依存する事になります。LaTeX には中途ファイルである (`file`).aux が使われています。LaTeX 自身もタイプセットを完全なものにするために (`file`).aux が変更されたかどうかを検査します。この (`file`).aux が前回のタイプセットから変更されていない場合はタイプセットが完了したとみなされます。ですから、この依存関係は次のようになります。

```
<file>.dvi: <file>.aux
<file>.aux: <file>.tex
—————→latex <file>
```

具体的には次のように記述します。

```

FILE=hoge
TEX=latex
all: $(FILE).dvi
$(FILE).dvi: $(FILE).aux
————→latex $(FILE)
$(FILE).aux: $(FILE).tex
————→latex $(FILE)

```

変数を使う場合は ‘\$’ を先頭にし変数名を丸括弧で括ります。上記のような例はちよつとうまく行きません。 *file*.dvi を作成するのに毎回タイプセットが行われます。これは *file*.log に相互参照に関する変更があるかどうかを調べるだけにするほうが良いでしょう。次のような Makefile を作成するともう少しうまく行きます。Makefile 中でシャープ ‘#’ 以降はコメントになります。

```

#主となる原稿
FILE=hoge
TEX=latex
REFGREP=grep "^LaTeX Warning: Label(s) may have changed."
all: $(FILE).dvi
$(FILE).dvi: $(FILE).aux
————→(while $(REFGREP) $ $(FILE).log; do $(TEX) $(FILE); done)
$(FILE).aux: $(FILE).tex
————→$(TEX) $(FILE)

```

Make で使われるファイルには ‘Makefile’ という決まった名前を付けます。この ‘Makefile’ をタイプセットしたい *file*.tex と同じ場所におきます。後は端末などから

```
$ make
```

とすると *file*.dvi が作成されるので 1 度やってみると良いでしょう。

原稿を作成しているときは *file*.aux とか *file*.log や *file*.toc などが中途ファイルなのでタイプセット後には必要ありませんから、これを一括で削除できると良いでしょう。

```

clean:
————→rm -f $(FILE).aux $(FILE).log $(FILE).toc $(FILE).tex~

```

Emacs などを使っているときは *file*.tex~ も削除してくれると有り難いでしょう。上記の 2 行を先程の Makefile に追加しましょう。

原稿が複数に分割されている場合は *file1*.tex, *file2*.tex, *file3*.tex, *file4*.tex など複数ありますからこれを次のように定義しておきます。


```
SRC=file1.tex file2.tex file3.tex file4.tex
```

これらのファイルが $\langle file \rangle$.aux に依存するというのは、次のように書く事ができます。

```
SRC=file1.tex file2.tex file3.tex file4.tex
$(FILE).aux: $(FILE).tex $(SRC)
—————>$(TEX) $(FILE)
```

参考文献データベースが複数ある場合も同じように行いますので、次のようにします。

```
REF=biblio1.bib biblio2.bib biblio3.bib
$(FILE).aux: $(FILE).tex $(SRC) $(REF)
—————>$(TEX) $(FILE)
```

最終的に成形したいファイルが DVI ファイル $\langle file \rangle$.dvi だけではなく PDF ファイル $\langle file \rangle$.pdf や PostScript ファイル $\langle file \rangle$.ps も作成するときは $\langle file \rangle$.dvi に依存させて、成形手順を書きます。普通 $\langle file \rangle$.pdf や $\langle file \rangle$.ps も $\langle file \rangle$.dvi を成形してから作成します。pdfL^AT_EX などを使っているならば $\langle file \rangle$.tex から $\langle file \rangle$.pdf にいきなり変換されますから、この場合は $\langle file \rangle$.pdf を $\langle file \rangle$.tex に依存させて適切な処理内容を記述します。最終的に $\langle file \rangle$.ps と $\langle file \rangle$.pdf を作成するならば以下のように記述できます。

```
TEX =platex
DVIPS =dvipsk -Pd1 -t a4 -o $(FILE).ps
DVIPDFM=dvipdfmx -f d1base14.map -p a4 -V 4 -z 5 -o $(FILE).pdf
all: $(FILE).pdf
$(FILE).pdf: $(FILE).dvi
—————>$(DVIPDFM) $(FILE)
$(FILE).ps: $(FILE).dvi
—————>$(DVIPS) $(FILE)
```

おまけとして関係するファイルを一つの TAR 書庫にまとめて gzip で圧縮する事を考えましょう。これには書庫を作成するための作業用のディレクトリ $\$(FILE)src/$ を作成すると簡単です。マクロパッケージや索引用のスタイルなどの他のファイルは‘OTHERS’に忘れずにまとめておきましょう。

```
#SRC=file1.tex file2.tex file3.tex file4.tex
#OTHERS=mymacro.sty
#REF=biblio1.bib biblio2.bib biblio3.bib
#FILE=main
tar: $(FILE).tex $(SRC) $(OTHERS) $(REF) Makefile
—————>mkdir -p $(FILE)src/
—————>cp $(FILE).tex $(SRC) $(OTHERS) $(REF) Makefile $(FILE)src/
—————>tar czf $(FILE)src.tgz $(FILE)src/
—————>rm -fr $(FILE)src
```

関係する全てのファイルを\$(file)src ディレクトリにコピーします。次にそのディレクトリに対して tar を使って\$(file)src.tar.gz を作成します。処理が終わったならば作業用のディレクトリの中を全て削除して完了です。こうすると\$(file)src.tar.gz がディレクトリ付きの書庫として作成されます。

以上のような事をレポート・論文作成時にも活用します。少々長いのですが以下の記述を Makefile にまとめます。

```
# Title: Makefile
# Date: 2004/03/28
# Name: Thor Watanabe
# Mail: thor@tex.dante.jp
# 主となる原稿
FILE→=sample
# 分割され、インクルードされているファイル
SRC→=#chap1.tex chap2.tex,..., chap<n>.tex
#スタイルファイルやクラスファイルなど
OHTERS→=#funthesis.cls mymacro.sty
# 画像などのバイナリファイル
IMG→=#hoge.eps capture1.jpg
#文献データベース, あるならば.
REF→=biblio.bib
#走らせる TeX プログラム
TEX=platex
# Red Hat の場合
#DVIPS→=pdvips -Ppdf
#XDVI→=pdxvi
# GNU Linux の場合
#DVIPS→=dvips -Ppdf
#XDVI→=xdvik
# Windows の場合
DVIPS→=dvipsk -Pdl -t a4
XDVI→=dviout
# dvipdfmx
DVIPDF =dvipdfmx -p a4 -f dlbase14.map -o $(FILE).pdf
# 相互参照の解消のため
REFGREP=grep "^LaTeX Warning: Label(s) may have changed."
# プリンタの設定
PRINTER=//server/printername
# 標準のターゲット
all:→$(FILE).ps $(FILE).pdf
printps: $(FILE).ps
→lpr -P$(PRINTER) $(FILE).ps
$(FILE).pdf: $(FILE).dvi
→$(DVIPDF) $(FILE)
$(FILE).ps: $(FILE).dvi
```

```

————→$(DVIPS) -o $(FILE).ps $(FILE)
$(FILE).dvi: $(FILE).aux $(FILE).bbl
————→(while $(REFGREP) $(FILE).log; do $(TEX) $(FILE); done)
$(FILE).bbl: $(REFFILE)
————→$(BIBTEX) $(FILE)
$(FILE).aux: $(FILE).tex
————→$(TEX) $(FILE)
clean:
————→rm -f $(FILE).aux $(FILE).log $(FILE).toc $(FILE).dvi
————→rm -f $(FILE).pdf $(FILE).tex~ $(FILE).lof $(FILE).lot
tar:
————→mkdir -p $(FILE)
————→cp $(SRC) $(OHTERS) $(REF) $(IMG) $(FILE).tex Makefile $(
FILE)/
————→tar czf $(FILE)src.tgz $(FILE)/
————→rm -fr $(FILE)/

```

適宜設定などに関してはご自分の環境やファイル状況に合わせてください。
レポートなどのそれ程大規模ではない文書の場合は次のようにします。

```

FILE=report
#SRC=file1.eps file2.eps
XDVI=xdvi
TEX=platex
all:—→$(FILE).dvi
$(FILE).dvi: $(FILE).tex
————→platex $(FILE) && platex $(FILE) && platx $(FILE)
$(FILE).pdf: $(FILE).dvi
————→dvi2pdf -p a4 -o $(FILE).pdf $(FILE)
clean:
————→rm $(FILE).aux $(FILE).log $(FILE).tex~
view:—→$(FILE).dvi
————→$(XDVI) $(FILE) &
tar:
————→mkdir -p $(FILE)src/
————→cp $(FILE).tex Makefile $(SRC) $(FILE)src/
————→tar czf $(FILE)src.tgz $(FILE)src/
————→rm -fr $(FILE)src/

```

上記のファイルを作成しておけば

```
$ make view
```

とすると report.tex の変更に合わせてタイプセットをしてから hoge.dvi を表示し、

```
$ make tar
```

とするとそのときのレポートに関連するファイルを `reportsrc.tgz` として書庫化してくれます。そのためにはあらかじめ関係ファイルを SRC に指定します。

```
SRC=file1.eps file2.eps
```

最近自分が良く使っている Makefile のサンプルを以下に示します*8。

```
# \begin{setting}[Suffixes]
# \comment{拡張子の登録}
.SUFFIXES: \
———→.aux .bb .bbl .bib .blg .bmp .cls .dic .dtx .dvi .eps \
———→.htm .html .idx .ilg .ind .ins .jpg .jpeg .lof .log .lol \
———→.lot .mp .obj .out .pdf .pl .plt .png .ps .sty .tex .toc
# \end{setting}

# \begin{setting}[Programs]
# \comment{DVI 生成}
TEX=——→platex
# \comment{索引生成}
MAKEIDX=mendex
# \comment{DVI \to PDF 生成}
DVIPDF= dvipdfmx
# \comment{削除}
RM=——→rm -f
# \comment{再起的削除}
RMFR=——→rm -fr
# \comment{再起的ディレクトリ作成}
MKDIR=——→mkdir -p
# \comment{Make の本体}
AUTOMAKE=make
# \end{setting}

# \begin{setting}[Filenames]
# \comment{基本となるファイル}
F=——→main
# \comment{基本となるマクロパッケージ}
M=——→mymacros
# \comment{索引スタイル}
IST=——→$M.ist
# \comment{必要となるソース}
SRC=——→$(IST) $F.dic $F.tex $F.pdf \
———→$M.dtx $M.ins $M.sty $M.pdf \
———→Makefile makebb
# \comment{画像のディレクトリ}
IMG=——→img
```

*8 <http://tex.dante.jp/jou1/Makefile>

```
# \end{setting}

# \begin{setting}[CommandLineOptions]
# \comment{$TEX のオプション}
TEXOPT= -src-specials
# \comment{$DVIPDF のオプション}
DVIPDFOPT=————→-O 2 -p a4 -V 4 -r 1200
# \comment{$DVIPS のオプション}
DVIPSOPT=————→-t a4
# \comment{$MAKEIDX のオプション}
IDXOPT= -s $(IST) -d $F.dic
# \end{setting}

# \begin{setting}[Printer]
# \comment{ラインプリンタ}
LPNAME= 192.168.11.16
# \end{setting}

# \begin{setting}[Programs]
# \begin{subsetting}[Mac OS X]
# \comment{DVI ファイルプレビューア}
XDVI=—→mxdvi
# \comment{PDF ファイルプレビューア}
PDFVIEWER= open
# \comment{DVI \to PS 生成}
DVIPS= dvips
# \end{subsetting}

# \begin{subsetting}[Linux]
#XDVI= xdvi # or pxdvi
#PDFVIEWER= acroread
#DVIPS= dvips # or pdvips
# \end{subsetting}

# \begin{subsetting}[Windows + Cygwin]
#XDVI= /cygdrive/c/dviout/dviout.exe
#PDFVIEWER= /cygdrive/c/Program\ Files/Adobe/Reader32.exe
#DVIPS= dvipsk
# \end{subsetting}
# \end{setting}

# \begin{dependency}
# \comment{標準のターゲット}
all: $F.pdf

$F.dvi: $F.tex
$F.ind: $F.idx
```

```

$F.idx: $F.tex
$F.pdf: $F.dvi

$M.dvi: $M.dtx
$M.pdf: $M.dvi
$M.sty: $M.dtx
# \end{dependency}

# \begin{GenerationRules}
.tex.dvi:
———→$(TEX) $*; $(TEX) $*;
# \comment{相互参照の解決が終了するまでタイプセットを続ける.}
———→(while egrep "may_have_changed" *.log; do $(TEX) $*; done)
.dtx.dvi:
———→$(TEX) $<; $(TEX) $<
.tex.idx:
———→$(TEX) $*;
.idx.ind:
———→$(MAKEIDX) $(IDXOPT) -o *.ind *.idx
.dvi.pdf:
———→$(DVIPDF) $(DVIPDFOPT) $*
.dvi.ps:
———→$(DVIPS) $(DVIPSOPT) -o $@ $*
.dtx.sty:
———→$(TEX) $*.ins
# \end{GenerationRules}

# \begin{aliases}
# \begin{subsetting}[Primary]
# \comment{$F.tex の PDF ファイル $F.pdf の作成}
pdf:→$F.pdf
# \comment{$F.tex の DVI ファイル $F.dvi の作成}
dvi:→$F.dvi
# \comment{$F.tex の索引 $F.ind の作成}
ind:→$F.ind
# \comment{マクロパッケージ istt.sty の作成}
sty:→$M.sty
# \comment{マクロパッケージ istt.sty のマニュアル}
man:→$M.pdf
# \end{subsetting}

# \begin{subsetting}[Secondary]
# \comment{依存関係を無視した二度のタイプセット}
fast:
———→$(TEX) $F; $(TEX) $F
# \comment{依存関係を無視した一度のタイプセット}
faster:

```

```

————→$(TEX) $F
# \comment{今日の日付の PDF ファイルを作成}
today:→$F.pdf
————→cp $< $F-'date +%Y-%m-%d-%H-%M'.pdf
# \end{subsetting}

# \begin{subsetting}[Previewing]
view:→$F.dvi
————→$(XDVI) $<
viewpdf: $F.pdf
————→$(PDFVIEWER) $<
# \end{subsetting}
# \end{aliases}

# \begin{setting}[Distribution]
clean:
————→$(RM) $F.aux $F.log $F.toc $F.dvi $F.ind
————→$(RM) $F.out $F.brj $F.idx $F.ilg *~
# \comment{原稿の配布用 TAR ball}
tar:→$M.pdf $F.pdf
————→$(MKDIR) $(F)src/img
————→cp -r img $(F)src/
————→cp $(SRC) $(F)src/
————→tar czf $(F)src.tgz $(F)src/
————→$(RMFR) $(F)src/
# \comment{アーカイブのチェック}
checktar:
————→tar tzf $(F)src.tgz | sort | less
# \comment{PDF のファイル情報とフォント情報のチェック}
checkpdf: $F.pdf
————→pdftools $< | less;
————→pdffonts $< | less
# \comment{画像 ($IMG ディレクトリ) のBoundingBox の作成}
bb:
————→./makebb $(IMG)
ls:
————→ls -lhGF --color=auto
# \end{setting}

```

これによりターゲットとして以下のものが指定できるようになります。

```

all  checkpdf  clean  fast    ind  man  sty  today  viewpdf
bb   checktar  dvi    faster  ls   pdf  tar  view

```

技術的な細かい部分についての説明は他の参考文献を参照して下さい。ここで重要な事は最低限ファイルとして `main.tex`, `makebb`, `mymacro.dtx`, `mymacro.ins`,

mymacro.ist がある事, img ディレクトリに画像が存在している事です.

▶ **問題 8.11** 実際に必要なファイルを揃えた簡単なサンプルファイルをウェブ上に用意しておきます*9. これを使って各々のターゲットがどのような役割を持っているのか, それを考えながらサンプルを実行し, その結果を吟味してください. 素の Makefile では環境に依存する記述が含まれているため, 適宜自分の環境に合わせて修正して下さい.

▼ 8.20.3 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, Perl スクリプトで書かれた John Collins 氏による latexmk というものも紹介しましょう. latexmk を日本語環境で動かせるために latexmk.pl に変更を加えます.

```
$latex = 'latex'; → $latex = 'platex';  
$bibtex = 'bibtex'; → $bibtex = 'jbibtex';  
$makeindex = 'makeindex'; → $makeindex = 'mendex';  
$dvips = 'dvips'; → $dvips = 'dviipsk';
```

あとは端末などから

```
$ latexmk file.tex
```

とすると標準では DVI ファイルが作成されます. BibT_EX 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.

▼ 8.20.4 原稿の版管理——CVS

例えば, ごく普通にテキストエディッタで文書を作成していたならば, 「過去の状態の復元」などのような作業は非常に困難でしょう. これを手動で行うにはテキストの文書であれば過去のファイルを更新したファイルとは別に保存しておかなければなりません. この作業は面倒なのでついつい上書きしがちです. しかし, 過去に削除してしまった項目が実は後で必要になるという事態はときたま起こるものです. このような問題を防ぐために版管理 (バージョンコントロール) というシステムが必要になります. 現在広く使われているのは Unix 系 OS でも Windows でも動作するような, プラットフォームに依

*9 <http://tex.dante.jp/jou1/mainsrc.tgz>

存しない版管理プログラムです。Brian Berliner 氏、David Zuhn 氏、Jim Kingdon 氏、Jeff Polk 氏らによる CVS や、GNU Emacs の VC などが有名です。CVS に関しては日本語化マニュアル^{*10}を参照してください。

8.21 簡単な計算と条件分岐

▼ 8.21.1 簡単な四則演算——calc

数値等の計算を行なうために、 \TeX には次の計算用の命令があります。

```
\advance<要素> by <数値>
\multiply<要素> by <数値>
\divide<要素> by <数値>
```

\TeX には次のような命令が用意されています。

```
\newcounter{(カウンタ名)}
\setcounter{(カウンタ名)}{<数値>}
\addtocounter{(カウンタ名)}{<数値>}
\newlength{(長さ変数)}
\setlength{(長さ変数)}{<伸縮長の設定>}
\addtolength{(長さ変数)}{<伸縮長の設定>}
```

この \setcounter や \addtocounter , \setlength , \addtolength を拡張して、計算式も対応させて拡張したのが Kresten Krab Thorup 氏と Frank Jensen 氏による calc パッケージです。他にも \LaTeX では要素の高さ、幅、深さを長さに代入する命令が拡張されています。

```
\settowidth{(長さ変数)}{<LR モードの要素>}
\settoheight{(長さ変数)}{<LR モードの要素>}
\settodepth{(長さ変数)}{<LR モードの要素>}
```

このようにして <長さ変数> に値を代入できます。

```
\setlength{\parskip}{0.68\mylength}
```

このようにしても良いのですが、もっと簡単に計算できれば便利です。

^{*10} <http://www.radiofly.to/nishi/cvs/>

```
\setlength{\parskip}{\widthof{$f + g$} * \real{0.68}}
```

実数値はちよつと特別ですから、`\real` 命令を補います。次のように `\widthof` の他にも次の命令が使用できます。

```
\widthof{<LR モードの要素>}
\heightof{<LR モードの要素>}
\depthof{<LR モードの要素>}
```

これにより次のような記述が可能です。

```
\usepackcalc}
\setcounter{uho}{3 + 5 * 4 - \value{page}}
\setlength{hoge}{\widthof{$f + g$} + \value{section}pt * \real
{2.5}}
```

▷ **例題 8.12** 以下の原稿の出力結果を吟味して下さい。

```
\documentclass{jsarticle}
\newcounter{tnumber}
\newcommand\Use{\setcounter{tnumber}{0}}
\newcommand\Num{\stepcounter{tnumber}\thetnumber}
\begin{document}
\begin{tabular}{clr}
\hline
\Use 番号 & 氏名 & 得点\\
\hline
\Num & Donald Knuth & 100\\
\Num & Leslie Lamport & 95\\
\Num & 未来 太郎 & 90\\
\Num & Th'or Watanabe & 85\\
\hline
\end{tabular}
\end{document}
```

これにより列の通し番号を自動的に振る事ができるようになるのが分かるでしょう。

▶ **問題 8.13** 例題 8.12 を改良して、さらに得点の平均点を算出するにはどうすれば良いでしょうか。L^AT_EX における算術は `calc` パッケージを使えば、記述が簡略化されます。

```
\documentclass{jsarticle}
\usepackage{calc}
```

```

\newcounter{tnumber} \newcounter{goukei} \newcounter{heikin}
\newcommand\Use{\setcounter{tnumber}{0}%
  \setcounter{heikin}{0}}
\newcommand\Num{\stepcounter{tnumber}\thetnumber}
\newcommand\Sum[1]{\addtocounter{goukei}{#1}#1}
\newcommand\Avr{\setcounter{heikin}{\value{goukei}/\value{tnumber}
  }}%
  \theheikin}
\begin{document}
\begin{tabular}{clr}
\hline
\Use 番号 & 氏名 & 得点\\
\hline
\Num & Donald Knuth & \Sum{100}\\
\Num & Leslie Lamport & \Sum{95}\\
\Num & 未来 太郎 & \Sum{90}\\
\Num & Th'or Watanabe & \Sum{85}\\
\hline
平均 & & \Avr \\
\hline
\end{tabular}
\end{document}

```

平均点の結果は '92' となります。余力があれば実数にも対応できるように改良してみてください。

▼ 8.21.2 条件分岐——ifthen

LT_εX には様々な条件分岐を実現するためのコマンドが用意されています。日常的に条件分岐を必要とするような方は David Carlisle 氏による ifthen パッケージを用いると良いと思います。

```

\ifthenelse{<条件文>}{<真の場合>}{<偽の場合>}
\whiledo{<条件文>}{<処理>}

```

ブール型変数に関するコマンドも追加されています。

```

\newboolean{<名前>}
\setboolean{<名前>}{<true|false>}

```

\ifthenelse, \whiledo における<条件文>としては長さの比較 (\lengthtest), 奇数かどうかの判定 (\isodd), 文字列の比較 (\equal) などがあります。

```
\boolean{<名前>} (ブール値を評価)
\isodd{<数値>} (奇数かどうか)
\equal{<文字列1>}{<文字列2>}
\lengthtest<<長さ1> <関係演算子> <長さ2>>
```

▶ **問題 8.14** 以下の原稿の出力結果を吟味して下さい。

```
\usepackage{ifthen}% プリアンブルで
\newcounter{test}% プリアンブルで
\newcommand*\Ax[3][a]{%
  \setcounter{test}{0}
  \whiledo{\value{test}<#2}{%
    \stepcounter{test}#1\sb{\thetest}#3}}
\begin{displaymath}
f = \Ax[\sqrt a]{10}{+}
  \Ax[\sqrt b]{9}{+} \sqrt b_{10}
\end{displaymath}
```

▷ **例題 8.15** 以下の入力例を実際にタイプセットし、その出力結果を吟味してください。

```
あれは\ifthenelse{\isodd{\pageref{are}}}{奇数}{偶数}ページにあります。
この文書は\ifthenelse{\boolean{draft}}{完成版}{未完成版}です。
```

結果は「あれは奇数ページにあります。この文書は完成版です」となるでしょう。

▷ **例題 8.16** 以下の入力例を実際にタイプセットし、その出力結果を吟味してください。

```
\def\A{未来} \def\B{未来}
ここは\ifthenelse{\equal{\A}{\B}}{北海道}{函館}\par
多分\ifthenelse{\lengthtest{\textwidth < \textheight}}{好き}{嫌い}だ
ろう
```

結果は「ここは北海道 多分好きだろう」となるでしょう。

▷ **例題 8.17** 以下の入力例を実際にタイプセットし、その出力結果を吟味してください。

```
\newboolean{hoge}
\ifthenelse{\boolean{hoge}}{true}{false}
\setboolean{hoge}{true}
\ifthenelse{\boolean{hoge}}{true}{false}
```

結果は ‘false true’ となります。

▷ **例題 8.18** 以下の入力例を実際にタイプセットし、その出力結果を吟味してください。

```
\newcounter{hoge}
\setcounter{hoge}{100}
\whiledo{\value{hoge}<107}{\roman{hoge}\space\stepcounter{hoge}}
```

結果は ‘c ci cii ciii civ cv cvii cvii’ となります。

`\and`, `\or`, `\not`, `\(`, `\)` を使うと複合条件判断もできます。

▶ **問題 8.19** 以下の記述を実際にタイプセットし、その出力結果を吟味してください。

```
\newboolean{A} \newboolean{B} % 入力 A/B
\newcommand*{\TF[1]{\ifthenelse{\boolean{#1}}{1}{0}}% 論理値を出す
\newcommand*\SET[2]{\setboolean{A}{#1}\setboolean{B}{#2}}% 入力 A/B
  の値の設定
\newcommand*\AND[2]{% 論理積
AND(\TF{A}, \TF{B}) $=$ \ifthenelse{\boolean{A}\and\boolean{B}
}{1}{0}}
\newcommand*\XOR[2]{% 排他的論理和
XOR(\TF{A}, \TF{B}) $=$ \ifthenelse{%
  \(\(\not \boolean{#1}\) \and \boolean{#2}\)\or
  \(\ \boolean{#1} \and \(\not \boolean{#2}\)\)}{1}{0}}
\SET{false}{false} \AND{A}{B}, \XOR{A}{B}\par
\SET{false}{true} \AND{A}{B}, \XOR{A}{B}\par
\SET{true}{false} \AND{A}{B}, \XOR{A}{B}\par
\SET{true}{true} \AND{A}{B}, \XOR{A}{B}\par
```

結果は次のようになります。

AND(0, 0) = 0, XOR(0,0) = 0

AND(0, 1) = 0, XOR(0,1) = 1

AND(1, 0) = 0, XOR(1,0) = 1

AND(1, 1) = 1, XOR(1,1) = 0



第 9 章

文書のサンプル

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。著者の大学の場合は `funthesis.cls` というファイル名でウェブページにて配布されています*1。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。

9.1 投稿・概要論文のサンプル

`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}
ここに概要を記述します。
\end{onecolabstract}}]
```

`jsarticle` を使った例のソースファイルです。「途中省略」とある部分は、テキストを省略しています。

```
\documentclass[twocolumn,papersize,dvipdfmx]{jsarticle}
\columnseprule 0.5pt% 段間の罫線
\usepackage{type1cm,epic,eepic,amssymb,amsmath,graphicx,url}
\title{2段組での投稿論文のサンプル}
\author{{\small システム情報科学部 情報アーキテクチャ学科}\
m1201234 函館 花子 \ 指導教員 未来 太郎}
\begin{document}% 本文開始
```

*1 著者が若干介入している時期がありました。

```
\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。途中省略。

```
\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。途中省略。

```
\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{tab:result}`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`\K{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]
```

```
\centering
```

```
\caption{データの集計結果}\label{tab:result}
```

```
\begin{tabular}{lrr}
```

```
\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 \\ \hline
```

```
合計 & 166 & 100 \\ \hline
```

```
\end{tabular}
```

```
\end{table}
```

途中省略。

```
\section{考察}
```

途中省略。

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*}
```

$$f(x) = \frac{1}{2} \left(\sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \right)$$

$$a_n = \frac{1}{L} \int_{-L}^L f(u) \cos \frac{n\pi u}{L} du$$

$$b_n = \frac{1}{L} \int_{-L}^L f(u) \sin \frac{n\pi u}{L} du$$

```
\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. \nonumber \\
& + & \left. \left. \frac{1}{L} \int_{-L}^L f(u) \sin \frac{n\pi u}{L} du \cdot \sin \frac{n\pi x}{L} \right] \right. \\
& & \left. \label{eq:fourier1} \right.
\end{eqnarray}

```

式 $\text{\ref{eq:fourier1}}$ を $(L \rightarrow \infty)$ にしたりしてフーリエ変換は一般に式 $\text{\ref{eq:fourier2}}$ 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 \label{eq:fourier2}
\end{equation}

```

式 $\text{\ref{eq:fourier2}}$ 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 $\text{\ref{fig:sample}}$ となる。

```

\begin{figure}[htbp]
\centering
\fbbox{\rule{0pt}{3zw}\rule{3zw}{0pt}}
\caption{picture 環境で描画した図形}\label{fig:sample}
\end{figure}
\nocite{*}
\bibliographystyle{jplain}
\bibliography{\jobname}% 参考文献にファイル名.bib を指定
\end{document}

```

2 段組での中間報告のサンプル

システム情報学部 情報アーキテクチャ学科

m1201234 函館 花子

指導教員 未来 太郎

2006 年 2 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、はこで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 データの集計結果

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。

この現象を天下降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。

$$\begin{aligned} f(x) &= \frac{1}{2L} \int_{-L}^L f(u) du \\ &+ \sum_{n=1}^{\infty} \left[\frac{1}{L} \int_{-L}^L f(u) \cos \frac{n\pi x}{L} du \cdot \cos \frac{n\pi x}{L} \right. \\ &\left. + \frac{1}{L} \int_{-L}^L f(u) \sin \frac{n\pi u}{L} du \cdot \sin \frac{n\pi x}{L} \right] \quad (1) \end{aligned}$$

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 \LaTeX コンパニオン. 東京アスキー, 1998.

- [2] 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.
- [3] 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.
- [4] 乙部敏己, 江口庄英. \LaTeX 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 \TeX ブック. アスキー出版局, 1992.

9.2 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` を使う事になると思います。

`jsbook` の場合にクラスオプションは次のようにすると左右起こしをせずに片面印刷で出力されます。

```
\documentclass[11pt,report]{jsbook}
```

`jreport` や `jsbook` で使用できる見出しは `\chapter`, `\section`, `\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次のようにして章立てします。

```
\chapter*{概要}\addcontentsline{toc}{chapter}{概要}
ここに簡潔に概要を書く。
```

▼ 9.2.1 クラスファイルが提供されている場合

学位論文などは大学側から文書クラスが提供される事があります。

```
%\documentclass[english]{funthesis}%本文が英語のとき
\documentclass{funthesis}
\usepackage[dvipdfmx]{graphicx}%dvips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}
どっこいどっこいです.
\chapter{結論と今後の展開}
どれもだめでした.
% 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}行\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}

```

▼ 9.2.2 クラスファイルが提供されていない場合

もし大学・機関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事になります。大抵の場合は 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るのが普通のようなようです。親切な方がすでに作成してウェブで公開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。まずはご自分の提出・投稿先の規定に合わせて jsbook の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。ファイルの先頭に次のようなファイル情報を書き込んでおくと良いでしょう。

```

%% File: mygs.sty
%% Copying : Your Name
%% E-mail : name@univ.ac.jp
%% Date : 2006/04/20
\ProvidesPackage{mygs}[2006/04/20 First Family]

```

大抵の機関では Times 系のフォントを指定すると思いますので次の 1 行もあると良いでしょう。

```

\RequirePackage{txfonts}

```

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。

次にページレイアウトについてです。マージンについても細かい指定をしてくるかもしれませんが、一定の設定方法を紹介しましょう。ページレイアウトで設定できる項目については図 8.1 を見てください。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 が必要
```

行数が 40 行と指定されているのであれば、`\textheight` に 40 行送り分の長さ (`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、ページレイアウトの設定をしても良いでしょう。

```
\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@}
```

ここでの `\p@` は単位 ‘pt’ の事です。マクロの中ではこのような命令を使うといいそうです。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` の場合はフッターの中央にページ番号を出力するようにします。

```
\pagestyle{plainfoot}
```

`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次のようにします。

```
\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
```

ページ番号を太字にするときは `\normalfont` の後に `\bfseries` を追加します。見出しのフォントで和文をゴシック、欧文は Times Bold にしたいとき、`jsbook` の場合は次のようにしておけば良いでしょう。

```
\renewcommand{\headfont}{\gtfamily\rmfamily\bfseries}
```

`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` も次のようにすると `\subsection` まで出力されます。

```
\setcounter{tocdepth}{2}
```

`jreport` の場合は見出しの後の字下げが行われない事がありますので次のようにして `indentfirst` パッケージを読み込みます。

```
\RequirePackage{indentfirst}
```

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*2。

```
%% File: mygs.sty
%% Copying : Thor Watanabe
%% E-mail : thor@tex.dante.jp
%% Date : 2006/04/20
\ProvidesPackage{mygs}[2006/04/20 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@}
```

*2 <http://tex.dante.jp/jou1/mygs.sty>


```

\setlength\footskip{2\baselineskip}%必要に応じて
\def\ps@foot{%フッターに ‘-- ページ番号 --’ としていたいとき
\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次のようにします。

```
\documentclass[a4j,11pt,report]{jsbook}
```

そして先程作成した `mysg.sty` を `\usepackage` 命令で読み込みます。

```
\usepackage{mysg}
```

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事もできます。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\vskip2\cvs%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\vskip\cvs
{\Large\normalfont 未来太郎}\par\vskip2\cvs
{\small 未来研究学科 \quad 学籍番号}\par\vskip.5\cvs
{\small 指導教員 \quad 北海太郎}\par\vskip2\cvs
{\提出日 2006/03/30}\par\vskip3\cvs
{\Large\headfont English Title}\par\vskip\cvs
{\large\rmfamily Your Name}\par\vskip\cvs
\end{center}%
\vfill\null

```

```
\end{titlepage}}
```

`\vskip` とは垂直方向に空き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。

見出しを出力する `\section` 命令の体裁が適切ではないと思われるのであれば、適宜 `\section` 型の命令を再定義します。

```
\@startsection{<見出しの種類>}{<階層>}{<字下げ>}{<前空き>}{<後空き>}{<体裁>}*{<目次用の見出し>}{<見出し>}
```

▷ **例題 9.1** 実際に `jsbook.cls` から `\section` の定義を何かしらのファイルにコピーしてください。

```
\makeatletter
\renewcommand{\section}{%
  \if@slide\clearpage\fi
  \@startsection{section}{1}{\z@}%
  {\Cvs \@plus.5\Cdp \@minus.2\Cdp}% 前アキ
  {.5\Cvs \@plus.3\Cdp}% 後アキ
  {\normalfont\Large\headfont\raggedright}}
\makeatother
```

`\newcommand` は `\renewcommand` に変更します。ここで、`\raggedright` となっている箇所を `\centering` 等に変更すると結果はどのようなようになるか吟味してください。

他にも章見出しの体裁は `\@makechapterhead`, `\@makeschapterhead`, 図表見出しの体裁は `\@makecaption`, 文献一覧のラベルの体裁は `\@biblabel` という具合に、それぞれ対応するコマンドが存在します。

▶ **問題 9.2** 次の記述を追加すると `\section` 等の命令はどのように変わるでしょうか。

```
\renewcommand{\secCNTformat}[1]{\@nameuse{the#1}.\quad}
```

実際にタイプセットし、出力結果を吟味してください。

▶ **問題 9.3** 章見出しは実際に `\@makechapterhead` 命令で表示されます。7 行目の `'\par\nobreak'` と `'\vskip \Cvs'` という記述を削除するとどうなりますか？ また、`\raggedright` を `\centering` 等に変更するとどのような出力結果になるのか、実際にタイプセットして確認してください。

```

\def\@makechapterhead#1{%
  \vspace*{2\Cvs}% 欧文は 50pt
  {\parindent \z@ \raggedright \normalfont
   \ifnum \c@secnumdepth >\m@ne
     \if@mainmatter
       \huge\headfont \@chapapp\thechapter\@chappos
       \par\nobreak
       \vskip \Cvs % 欧文は 20pt
     \fi
   \fi
   \interlinepenalty\@M
   \Huge \headfont #1\par\nobreak
   \vskip 3\Cvs}} % 欧文は 40pt

```

▶ **問題 9.4** jsclasses の quotation 環境では、右余白が 0 pt に設定されています。それでは quotation 環境を下記のように再定義すると、結果はどのようになるのか、実際にタイプセットして確認してください。

```

\renewenvironment{quotation}{%
  \list{}{%
    \listparindent\parindent
    \itemindent\listparindent
    \rightmargin = \leftmargin}%
  \item\relax}{\endlist}

```

▶ **問題 9.5** 図表見出しは \@makecaption によって表示されています。では、‘\advance\leftskip1cm’ と ‘\advance\rightskip1cm’ という記述を削除すると結果はどうなるか、実際にタイプセットして確認してください。

```

\long\def\@makecaption#1#2{{\small
  \advance\leftskip1cm
  \advance\rightskip1cm
  \vskip\abovcaptionskip
  \sbox\@tempboxa{#1\hskip1zw\relax #2}%
  \ifdim \wd\@tempboxa <\hsize \centering \fi
  #1\hskip1zw\relax #2\par
  \vskip\belowcaptionskip}}

```

▶ **問題 9.6** 目次中における章見出しは \l@chapter で表示されます。では \begingroup の後に \large を記述するとどうなるか、実際にタイプセットし、それを確認してください。

```
\renewcommand*{\l@chapter}[2]{%
  \ifnum \c@tocdepth > \m@ne
    \addpenalty{-\@highpenalty}%
    \addvspace{1.0em \@plus\p@}
    \begingroup
      \parindent\z@
      \rightskip\@tocrmarg
      \parfillskip-\rightskip
      \leavevmode \large \headfont % ここに \large を追加
      \setlength\@lnumwidth{4.683zw}%
      \advance\leftskip\@lnumwidth \hskip-\leftskip
      #1\nobreak\hfil\nobreak\hbox to\@pnumwidth{\hss#2}\par
      \penalty\@highpenalty
    \endgroup
  \fi}
```

付録 A

最近の動向

$\text{T}_{\text{E}}\text{X}$ の世界も熱狂的な方々が各々の改良や研究をされているので、日々進歩しています。それらの開発・発展を見逃していると、せっかく便利なプログラムやパッケージが公開されているがもったいない事になりかねません。ですから、このページでは主に 2006 年 4 月現在 $\text{T}_{\text{E}}\text{X}$ 周辺で発展している便利なツールやパッケージを紹介しします。

A.1 PDF と $\text{T}_{\text{E}}\text{X}$

$\text{T}_{\text{E}}\text{X}$ というのは Donald Knuth 氏 という計算機科学者が何十年も前に開発したプログラムですので、幾分時代にそぐわない部分があると思います。そこで $\text{T}_{\text{E}}\text{X}$ を改良して $\varepsilon\text{-T}_{\text{E}}\text{X}$ なるものが開発されています。 $\text{T}_{\text{E}}\text{X}$ のレジスタ数を増やしたり、色々と新しいコマンドを追加していたりと便利なのですが、2006 年 4 月現在日本語化されていません^{*1}。

さらに $\text{T}_{\text{E}}\text{X}$ から直接 PDF ファイルを作成したいというのが希望としてあるのですが、実際に Hàn Thé Thành 氏らによる $\text{pdfT}_{\text{E}}\text{X}$ や $\text{pdfL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ というプログラムが存在します。これはフォントメトリクスと実フォント（または仮想フォント）の両方にアクセスする事で一気に PDF を作成するというものです。2006 年 4 月現在日本語化されていません。

さらに $\varepsilon\text{-T}_{\text{E}}\text{X}$ と $\text{pdfT}_{\text{E}}\text{X}$ をマージして $\text{pdf}\varepsilon\text{-T}_{\text{E}}\text{X}$ というのも生まれています。もちろん $\text{pdf}\varepsilon\text{-L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ もあります。近い将来 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\varepsilon}$ の後継バージョンである $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 3$ も登場するでしょうし、 $\varepsilon\text{-T}_{\text{E}}\text{X}/\text{pdfT}_{\text{E}}\text{X}$ が日本語化される日も近いと思われます。

Mac OS X の環境に依存しますが、 $\text{pdf}\varepsilon\text{-T}_{\text{E}}\text{X}$ をベースにして $\text{X}_{\text{U}}\text{T}_{\text{E}}\text{X}$ ^{*2} というプログラムもあります。これは Mac OS X の ATSUI: Apple Type Services for Unicode Imaging に直接アクセスし、システムのフォントを利用できるようになるものです。

^{*1} いくつかのアプローチで日本語を扱う試みはすでにありますが、実用段階にまで達していないのが現状です。

^{*2} <http://scripts.sil.org/xetex/>

A.2 文字と書体

TeX では標準的な Donald Knuth 氏による Computer Modern フォントのみならず、様々な書体を使う事ができるようになってきています。Computer Modern フォントを PostScript Type 1 形式で PDF や PostScript に埋め込みできる `type1cm` パッケージがあります。さらにヨーロッパ語圏のアクセント記号も含む `type1ec` パッケージも有用です。Times 系の書体 (Word の標準でもある) を本文に使いたいならば Young Ryu 氏による `txfonts`、Palatino 系の書体ならば `pxfonts` 等のパッケージが便利です (図 A.1)。

表 A.1 フォント関連のパッケージ一覧

パッケージ	ローマン	サンセリフ	タイプライタ	数式
指定なし	CM Roman	CM Sans Serif	CM Typewriter	CM Roman
<code>pxfonts</code>	Palatino 風	Helvetica 風	Monospaced 風	Palatino 風
<code>txfonts</code>	Times 風	Helvetica 風	Monospaced 風	Times 風
<code>lmodern</code>	LM Roman	LM Sans Serif	LM Typewriter	
<code>type1cm</code>	CM Roman	CM Sans Serif	CM Typewriter	
<code>type1ec</code>	EC Roman	EC Sans Serif	EC Typewriter	
<code>mathptmx</code>	Times			Times
<code>mathpazo</code>	Palatino			Palatino
<code>helvet</code>		Helvetica		
<code>avant</code>		Avant Garde		
<code>courier</code>			Courier	
<code>bookman</code>	Bookman	Avant Garde	Courier	
<code>newcent</code>	New Century Schoolbook	Avant Garde	Courier	
<code>charter</code>	Charter			

`lmodern` パッケージの中身を確認してみると、実際に `lmodern.sty` の中では次のように標準のファミリーが再定義されています。

```
\ProvidesPackage{lmodern}[2005/02/28]
\renewcommand{\rmdefault}{lmr}
\renewcommand{\sfdefault}{lms}
\renewcommand{\ttdefault}{lmtt}
\endinput
```

$$\begin{aligned}
 f(z) &= \frac{1}{2\pi i} \oint_C \sum_{n=0}^{\infty} \frac{(z-a)^n}{(z_0-a)^{n+1}} f(z_0) dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} \frac{n!}{2\pi i} \oint_C \frac{f(z_0)}{(z_0-a)^{n+1}} dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} f^{(n)}(a)
 \end{aligned}$$

type1cm

$$\begin{aligned}
 f(z) &= \frac{1}{2\pi i} \oint_C \sum_{n=0}^{\infty} \frac{(z-a)^n}{(z_0-a)^{n+1}} f(z_0) dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} \frac{n!}{2\pi i} \oint_C \frac{f(z_0)}{(z_0-a)^{n+1}} dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} f^{(n)}(a)
 \end{aligned}$$

mathpazo

$$\begin{aligned}
 f(z) &= \frac{1}{2\pi i} \oint_C \sum_{n=0}^{\infty} \frac{(z-a)^n}{(z_0-a)^{n+1}} f(z_0) dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} \frac{n!}{2\pi i} \oint_C \frac{f(z_0)}{(z_0-a)^{n+1}} dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} f^{(n)}(a)
 \end{aligned}$$

mathptmx

$$\begin{aligned}
 f(z) &= \frac{1}{2\pi i} \oint_C \sum_{n=0}^{\infty} \frac{(z-a)^n}{(z_0-a)^{n+1}} f(z_0) dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} \frac{n!}{2\pi i} \oint_C \frac{f(z_0)}{(z_0-a)^{n+1}} dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} f^{(n)}(a)
 \end{aligned}$$

pxfonts

$$\begin{aligned}
 f(z) &= \frac{1}{2\pi i} \oint_C \sum_{n=0}^{\infty} \frac{(z-a)^n}{(z_0-a)^{n+1}} f(z_0) dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} \frac{n!}{2\pi i} \oint_C \frac{f(z_0)}{(z_0-a)^{n+1}} dz_0 \\
 &= \sum_{n=0}^{\infty} \frac{(z-a)^n}{n!} f^{(n)}(a)
 \end{aligned}$$

txfonts

図 A.1 基本書体の変更例

文書に使われる基本書体を変更するには、基本的には既存のパッケージを読み込むか、もしくは宣言されているファミリーの設定を変更します。

例えば、Computer Modern フォントを基本的に使う場合は次のように三つの命令を再定義します。

```
\renewcommand \rmdefault {cmr}
\renewcommand \sfdefault {cmss}
\renewcommand \ttdefault {cmtt}
\normalfont % おまじない
```

Times, Helvetica, Courier を基本的に使うのであれば次のようにします。

```
\renewcommand \rmdefault {ptm}% Postscript TiMes
\renewcommand \sfdefault {phv}% Postscript HelVetica
\renewcommand \ttdefault {pcr}% Postscript CouRier
\normalfont % おまじない
```

実際にパッケージを使う時、typelec パッケージを使うには次のようにするのが良いでしょう。

```
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{typelec}
```

fontenc パッケージと textcomp はおまじない的に記述した方が良いでしょう。

▷ **例題 A.1** 既存の原稿に以下の記述を追加し、タイプセットの実行結果を吟味してください。

```
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{typelec}
```

また、`\usepackage{typelec}` となっている箇所を、`type1cm`, `lmodern`, `txfonts`, `pxfonts` 等でも試してみてください。

helvet は Times 等に比べて若干大きいので、`scaled` オプションで拡大縮小すると良いでしょう。

```
\usepackage[scaled=.92]{helvet}
```

単に `scaled` オプションだけを記述した場合は 0.95 倍されます。

```
\usepackage[scaled]{helvet} % 標準は 0.95 倍
```

▷ **例題 A.2** 以下のファイルをタイプセットし、実行結果を吟味してください。

```
\documentclass{jsarticle}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
```



```

\usepackage{lmodern}%
\makeatletter
\newcommand*{\showfont[1]{\texttt{\string#1} $=$ {\ttfamily #1}\par}
\newcommand*{\sampletext[1]{\texttt{\string#1} $=$
  {#1 This is a sample text.}\par}
\newcommand*{\showfontinfo{%
  \showfont \rmdefault \showfont \sfdefault
  \showfont \ttdefault \showfont \encodingdefault
  \showfont \familydefault \showfont \seriesdefault
  \showfont \shapedefault \sampletext \rmfamily
  \sampletext \sffamily \sampletext \ttfamily
  {\int^{\beta}_{\alpha} f(x)dx = \left[ g(x)\right]^{\beta}_{\alpha}}}}
\makeatother
\begin{document}
\showfontinfo
\end{document}

```

さらに `\usepackage` の `lmodern` を `pxfonts` や `txfonts` にして試してみてください。

- ▶ **問題 A.3** Palatino, Helvetica, Courier を基本的に用いるような設定は次のようになります。実際にタイプセットし、その出力結果を吟味してください。

```

\usepackage{mathpazo}% palatino
\usepackage[scaled]{helvet}% Helvetica
\usepackage{courier}% Courier

```

この設定と `pxfonts` パッケージを使った場合の出力はどのように異なるのか、実際にタイプセットして確認してみてください。

- ▶ **問題 A.4** Times, Helvetica, Courier を基本的に用いるようにする設定は次のようになります。

```

\usepackage{mathptmx}% Times
\usepackage[scaled]{helvet}% Helvetica
\usepackage{courier}% Courier

```

この設定と `txfonts` パッケージを使った場合の出力はどのように異なるのか、実際にタイプセットして確認してみてください。



Donald Knuth 氏が作成した（芸術作品である）Computer Modern フォントの完成度は非常に高く、あらゆるケースを想定して設計されています。詳しくは『好き好き L^AT_EX 2_ε 書体編』で解説する事になると思いますが、どのような配慮がされているのか、その片割れだけでも紹介しておきます。まずは以下のファイル `cmtest.tex` をタイプセットし、`Dvipdfmx` 等で PDF として作成し、`cmtest.pdf` のフォント情報を吟味してください。

```

\documentclass[a4j,papersize,english]{jsarticle}
\usepackage{type1cm}
\author{A. U. Th\'}or}

```

```

\title{A Short Story}
\date{\today}
\begin{document}
\maketitle
\tableofcontents
\section{A Headline}
\subsection{Next Headline}
This is a sample text\footnote{This
is a sample footnote.}.
\end{document}

```

コンソールから `pdfinfo cmtest.pdf` とすれば、フォント情報が表示される事になります。これだけ単純なファイルですが、使われている書体は少なくとも `cmr6`, `cmr7`, `cmr8`, `cmr10`, `cmr12`, `cmr17`, `cmss10`, `cmss12` の七つはあります^{*3}。

この事実から考えられる事は、少なくとも `type1cm` パッケージを使っているファイルでは ‘Computer Modern’ の Roman 体が 6, 7, 8, 10, 12, 17 pt 毎に用意されていて、それがサイズ毎に使われる場所が違うという事です。

実際に `type1cm` パッケージには次のような記述が存在します。

```

\DeclareFontShape{OT1}{cmr}{m}{n}{%
<-6>cmr5 <6-7>cmr6 <7-8>cmr7
<8-9>cmr8 <9-10>cmr9 <10-12> cmr10
<12-17> cmr12 <17-> cmr17}{%

```

現在、ポピュラーに使用されている PostScript フォントは、サイズ毎に適切な書体が選ばれるようにデザインされていない場合も多いでしょう。

▼ A.2.1 日本語とユニコード周辺

`pTeX`/`ɛTeX` は標準的には JIS X 0208 (JIS 基本漢字) までの文字集合しか扱う事ができません。この問題に関しては齋藤修三郎氏による UTF パッケージで対処できます。UTF ではユニコード 文字集合まで扱う事ができます。さらに Adobe-Japan1-6 までの文字集合に対応した OTF パッケージも開発されています。

`TeX` を拡張して多言語組版を可能にする試みとしては John Plaice 氏と Yannis Haralambous 氏による Omega, `ɛTeX` 用では Lambda というのがあります。この後継としては `e-TeX` をベースとした Aleph と、`ɛTeX` 用の Lamed 等がありますが、2006 年 4 月現在開発途中のシステムです。

^{*3} この他に和文書体の `GothicBBB-Medium` と `Ryumin-Light` の二つが使われています。

A.3 日本語クラスファイル

最近までは ASCII が日本語化した pTeX に同封されている `jarticle`, `jreport`, `jbook` を使っていたのですが、現在は奥村晴彦氏が管理されている `jsclasses` を使うのが良いでしょう*4。これには `jsarticle`, `jsbook`, `okumacro`, `okuverb`, `morisawa` などのクラスとマクロが同封されています。レポートや論文を作成する上でもこれらのクラス・マクロは非常に完成度が高いため、標準的に `jsclasses` を使う事を強く推奨します。

jsarticle `jarticle` の代用となるもの。 `english` オプションを付ける事で、欧文組の時の行送りになる。その他多くの改良点がある。

jsbook `jbook` の代わりとなるもので書籍や論文作成用のクラスとして用いる。 `report` オプションで `jreport` の代用となる。

okuverb `verbatim` 環境をちょっと華麗にするためのパッケージ。

okumacro 奥村氏が美文書作成入門等の著書を執筆するために必要になったマクロを集めたもの。

morisawa モリサワ基本 5 書体パッケージを使うためのマクロ。フォント選択については奥村氏の考え（好み）が入っているので、和欧文の書体選択の相性などに関してこだわりがある場合はそれぞれカスタマイズする必要があるでしょう。

ただし、クラスファイルというものは多少なりと製作者の好み等により体裁が調整されている場合がありますので、自分の求める体裁と差異がある場合は、適宜該当する箇所を修正する事になると思われます。

A.4 画像やグラフィックス周辺

近年まで画像は EPS 形式しか受け付けられないようなデバイスドライバがりましたが、今では PDF (EPDF) を直接扱う事ができる `Dvipdfm`、その後継の `Dvipdfmx` もありますので、状況はかなり変わっています。2006 年 4 月現在の状況を考えますと、日本語環境では `Dvipdfmx` を使うのが最良だと思われます。BMP, PNG, JPEG, PDF, EPS 形式の画像の張り込みに対応しています。

A.5 今後について

TeX は文字組版に関しては相当優秀なシステムであり、そのハイフネーションアルゴリズム、プログラムの並列化と最適性、処理速度、行分割、ページ分割、フォントシステムなどにおいては、現存する一般的な組版システムに負けない高品質な機能を実装しています。ただし、画像の扱い等に関連した部分はほとんど実装されていないため、外部のプログラムに依存しているのが現状です。今後も TeX とその周辺は改良・発展が続くと予想されますので、その周辺情報に関

*4 私 5 回ほどバグ取りを行っているので、ほぼ完成形に近いです。

してはサポートページ^{*5}を参照してください。

▼ A.5.1 ptetex

土村展之氏の功績のお陰で非常に便利な日本語 TeX デистриビューションがリリースされています。次期 Vine Linux (4.0) の TeX 環境として採用される話もあります。いままでは日本語 TeX 環境を整備したければ teTeX と呼ばれるデистриビューションに日本語化パッチを多数適用するという煩雑な作業を伴いました。しかし、土村氏の ptetex により日本語化された TeX, xdvi, dvips, Dvipdfmx 等を比較的簡単にインストールすることができます。

デистриビューションの違い等による詳しいインストール方法は土村氏のウェブページをご覧ください。ここでは Vine Linux を例に環境構築の例を示します。

現在 Linux で Adobe-Japan-1-6 程度まで対応している PDF ビューアは Adobe Reader 7.0 位だと思いますので、最新版の Adobe Reader をインストールしておきます。Adobe Reader に依存している OpenLDAP^{*6}をインストールしておきます。そして Adobe 社のサイトから Adobe Reader の Linux 用の最新版 RPM をダウンロードし、rpm コマンドでインストールします。

```
# apt-get install openldap openldap-devel
# rpm -ivh AdobeReader_jpn-7.0.5-1.i386.rpm
```

土村氏の ptetex は OS 付属の TeX と共存するのは避けた方が良いと思われますので、‘tetex’のパッケージを削除します。

```
# apt-get --purge remove tetex
```

さらに ptetex に依存する不足パッケージを以下のようにしてインストールします。

```
# apt-get install build-essential bison flex ed
# apt-get install zlib-devel libpng-devel ncurses-devel
# apt-get install XOrg-devel openMotif-devel
```

続いて土村氏がすでに用意してくださっている RPM 版^{*7}を入手します。date=20060330という部分は適宜変更してください。

```
# baseurl=http://tutimura.ath.cx/~nob/tex/ptetex/ptetex3/rpm
# date=20060330; ver=3.0
# wget $baseurl/Vine3-ptetex3-$date-1.i386.rpm.bz2
# wget $baseurl/tetex-texmf-$ver-1.noarch.rpm.bz2
```

アーカイブを取得した後に、bunzip2 (または bzip2 -d) でファイルを解凍し、rpm コマンドで RPM を (アップグレード) インストールします。

```
# bunzip2 *.bz2
# rpm -Uvh tetex-texmf-$ver-1.noarch.rpm
# rpm -Uvh Vine3-ptetex3-$date-1.i386.rpm
```

これで相当フレッシュな日本語 TeX 環境を構築する事が可能です^{*8}。

^{*5} <http://tex.dante.jp/typo/>

^{*6} <http://www.openldap.org/>

^{*7} <http://tutimura.ath.cx/~nob/tex/ptetex/ptetex3/rpm/>

^{*8} Vine Linux の場合、OS 付属の teTeX を削除したときに R-devel, bibtex2html, dvipdfmx, dvipng, hevea, jvf, latex2html, task-tetex, task-texmacro-info, tetex, tetex-doc, tetex-extra, tetex-macros, texmacro-otf, xdvi, xdvi-search 等が削除される事になる

```

\ifnum 42146=\euc"A4A2 %"
\AtBeginDvi{\special{pdf:tounicode EUC-UCS2}}\else
\AtBeginDvi{\special{pdf:tounicode 90ms-RKSJ-UCS2}} \fi
\documentclass[dvipdfm]{jsarticle}[2006/01/04]
\usepackage[url][2004/03/15]
\usepackage[type1cm][2002/09/05]
\usepackage{okumacro}[2004/08/23]
\usepackage[deluxe]{otf}[2004/08/17]
\usepackage{hyperref}[2002/06/06]
\usepackage{graphicx}[1999/02/16]
\usepackage{color}[1999/02/16]
\begin{document}
\title{ptetex は素晴らしい!!!}
\author{名無し権兵衛}
\date \today
\maketitle
\tableofcontents
\section{ptetex3 は素晴らしい!!!!}
土村氏\footnote{\url{http://www.nn.iij4u.or.jp/~tutimura/}}
のお陰で日本語 \TeX 環境を比較的簡単に構築する事ができるよう
になりました。本当にありがとう!
\section{otf のテスト}
土吉：\CID{13706}野屋\par
梯子高：\UTF{9AD9}島屋
\section{otf のテスト その2}
フェスティバル\CID{20654}\par
番組\CID{20556}\par
\CID{15728} キーを押す。 \par
\CID{16314} を心がけよう。
\end{document}

```

以上のサンプルを *(file).tex* として^{*9}, `platex file` とすれば *(file).dvi* が生成されます。これを `xdvi file` とすると、**FT2: Open Font Error** というエラーを表示する事になると思われます^{*10}。

これにより `xdvi` ではなく `Dvipdfmx` などで PDF に変換してプレビューしてみます。

```

$ dvipdfmx <file>
$ xpdf <file>.pdf

```

まずは `Xpdf` でプレビューしてみます。OTF パッケージのいくつかの文字が表示されていないようです^{*11}。

そこで `acroread hoge.pdf &` 等として Adobe Reader でプレビューしてみます。これにより

ため、場合によっては依存関係を見逃してパッケージを入れ直す事になるかもしれません。

*9 奥村晴彦氏のサンプルを参考にしています。

*10 FreeType2 ライブラリのエラーだと思われます。

*11 OTF パッケージに `expert` オプションを付けると、さらにプレビューで問題が起きる事でしょう。

先ほどまで表示されていなかったグリフも表示できているようです。Vine Linux に含まれている `xpdfopen` というパッケージを `apt-get install xpdfopen` としてインストールすれば、

```
$ F=file
$ pdfcolose --file $F.pdf
$ platex $F; dvipdfmx $F
$ pdfopen --file $F.pdf
```

とすれば、Adobe Reader の PDF ファイルを自動的に閉じ、タイプセット後に再びファイルを開くようになります^{*12}。上記のような処理を必要に応じてシェルスクリプトとして作成しておけば便利です。

A.6 環境依存の話

▼ A.6.1 Vine Linux

TeX とその周辺の使い勝手を考えれば、2006 年 4 月現在において自分は Vine Linux が最適だと感じております^{*13}。今後他にも良いディストリビューションが登場する可能性があると思いますので、最近の動向に目を向けてみてください。

基本的に Vine Linux の場合はコンソールツール APT でも、GUI の Synaptic からでもソフトウェアのインストールや削除、更新が可能です。コンソールの場合は管理者権限で `apt-get update` としてから

```
# apt-get install <パッケージ名>
```

とするだけでパッケージ化されたソフトウェアを導入可能です。新規にソフトウェアをインストールするときに、そのソフトウェアに依存したソフトウェアの導入も迫られる場合がありますので、その場合は `y` キー等を押してインストール作業を進めてください。すでに導入されているソフトウェアを更新するには

```
# apt-get update
# apt-get upgrade
```

の 2 行を打ち込むだけで終わりです。

以下に TeX とその周辺に関連するパッケージを紹介します。基本的なソフトウェアを導入したければコンソールから管理者権限で `apt-get install task-tetex` とするだけです。

TeX に関連した何らかのソフトウェアをインストールした後は `texhash` コマンドをおまじないとして管理者権限で `texhash` とすると、`mktextlsr` が実行され `ls-R` ファイルが更新されます。

task-tetex `teTeX` をまとめてインストールするためのパッケージ。 `jvf`, `tetex`, `tetex-extra`, `xdvik`, `dvipdfmx`, `VFlib`, `freetype` の七つのパッケージが主にインストールされます。

task-texmacro-info 情報工学に関する `teTeX` マクロがまとめてインストールされるパッケー

^{*12} DVI ファイルを `xdvi` でプレビューしていてもファイルを一度閉じてからタイプセットする必要はありませんでしたが、PDF の場合はファイルを一度閉じてからタイプセットする必要があるため、PDF をプレビューしながらの作業に煩雑な作業が必要でした。この `xpdfopen` があれば PDF ファイルでプレビューしながらタイプセットができるようになるでしょう。日本以外では `pdfTeX` 用に使われている場合が多いようです。

^{*13} 2006 年 4 月現在の Vine Linux の正式リリースは 3.2 となっています。

ジ. texmacro-his, texmacro-ieice, texmacro-ipsj の三つがインストールされます。

task-texmacro-phys 物理学に関する TeX マクロをまとめてインストールするためのパッケージ。ジ. task-tetex, texmacro-jps の二つが主にインストールされます。

tetex Thomas Esser 氏による TeX ディストリビューション TeX 。

tetex-doc TeX のマニュアルや周辺文書。 /usr/share/texmf/doc/ 以下に展開され、展開後のファイルサイズが 60 MB 程度と非常に大きい。しかし、 TeX のドキュメントを表示する texdoc コマンドにより `texdoc verbatim` とすると verbatim パッケージのマニュアルが自動的に開かれるため大変便利です。

tetex-extra TeX 関連の追加ソフトウェアとフォント。 txfonts, pxfonts 等がインストールされます。

tetex-macos TeX で使うマクロパッケージ集。 jsclasses-041229, prosper-1.00.4, kanjifonts-3.0, epsbox.sty, eclepsf.sty の五つがインストールされます。

xdvik Paul Vojta 氏らによる DVI プレビューアー xdvik の日本語化済プログラム。コンソールからは xdvi というコマンド名で実行できます。

各種学会のマクロパッケージ等も簡単に導入できます。マニュアル等は以下のディレクトリにあります。

/usr/share/doc/texmacro-〈パッケージ名〉-〈バージョン〉/

texmacro-his ヒューマンインタフェース学会論文原稿作成用マクロパッケージ。和文は his, 欧文は ehis パッケージを指定します。

texmacro-ieice 電子情報通信学会用マクロパッケージ。和文は ieicej, 欧文は ieice クラスを指定します。

texmacro-ipsj 情報処理学会論文原稿作成用マクロパッケージ。

texmacro-jps 日本物理学会論文原稿作成用マクロパッケージ。 jpsj2 クラスを指定します。

texmacro-otf 齋藤修三郎氏による OpenType Font 用の仮想フォントとマクロ。

この他にも IEEE, 人工知能学会, ソフトウェア科学会, 認知科学会等のウェブページで TeX クラス・マクロを公開しています。

その他便利だと思われるパッケージを紹介します。

書体 追加で以下の書体も導入可能です。

mathabx TeX 用の新しい数式フォント。

ec-fonts-mftraced Jörg Knappen 氏による Type1 PostScript 形式の EC フォント。原稿中で type1ec パッケージを指定すると使用できます。

tipa 福井玲氏による TeX 用の国際音標文字 (IPA: International Phonetic Alphabet) フォント TIPa。

Emacs (GNU) Emacs は Unix 系 OS で伝統的に使われているテキストエディタです。以下の拡張機能等を導入しても損はありません。

emacs テキストエディタ Emacs。

color-mate Emacsen 上でナイスなカラー表示をするための elisp。

yatex 広瀬雄二氏による Emacs 用の TeX 執筆支援環境 野鳥 (Ya TeX)。

xdvik-search TeX src-special を Emacsen 上で使用するための elisp。

エディタ 単なるエディタや WYSIWYG のワープロソフトもあります。

winefish \LaTeX 用のテキストエディタ.

lyx \LaTeX 形式でも保存できる簡易ワープロ LyX.

TeXmacs WYSIWYG ワープロ \TeX macs.

PDF 以下の PDF 関連のツールも導入しておくくと便利です.

dvipdfmx DVI \leftrightarrow PDF 変換 Dvipdfmx.

xpdf X ウィンドウシステム用 PDF ファイルビューア Xpdf.

pdftk Sid Steward 氏による PDF 操作ツールキット PDFtk.

ps2jpdf 漢字を埋め込まずに PostScript \leftrightarrow PDF に変換する ps2jpdf.

xpdfopen Adobe Reader に [ファイル] を [開く] / [閉じる] コマンドを送信.

BibTeX BibTeX 関連のツールです.

bibtex2html Jean-Christophe Filliâtre 氏と Claude Marché氏による (文献一覧).bib \leftrightarrow HTML 変換 BibTeX2HTML.

bibcheck Nelson Beebe 氏による BibTeX ファイルの整合性をチェック.

bibclean Nelson Beebe 氏による BibTeX ファイルの文法チェックと自動整形ツール.

bibutils 文献データ変換ユーティリティ. bib2xml, xml2bib 等がインストールされる.

ファイル変換 「(入力元) \leftrightarrow (出力先)」系のツールです.

detex Daniel Trinkle 氏による原稿から \TeX コマンドを取り除くプログラム DeTeX.

dvipng Jan-Ake Larsson 氏による DVI \leftrightarrow PNG へ変換.

latex2rtf \LaTeX \leftrightarrow RTF 形式に変換 \LaTeX 2RTF.

pstoedit PostScript/PDF \leftrightarrow 多種多様なベクトル形式に変換.

latex2html Nikos Drakos 氏による \LaTeX \leftrightarrow HTML 形式に変換するツール \LaTeX 2HTML.

科学系 数学や、科学分野で活躍するツールです.

ngraph 石坂智氏による 2次元グラフ作成プログラム Ngraph.

octave John Eaton 氏による行列演算を得意とする数値演算プログラム Octave.

scilab INRIA (フランス国立コンピュータ科学・制御研究所) による制御系を得意とする数値演算プログラム SciLab.

R 統計解析を得意とする数値演算プログラム R.

gnuplot Thomas Williams 氏と Colin Kelley 氏らによるグラフ描画プログラム Gnuplot.

画像/閲覧 画像編集, 閲覧, 加工を行うツールです.

gimp GNU の画像加工プログラム GIMP.

dia Alexander Larsson 氏らによる GTK+ ベースのダイアグラム作成プログラム Dia.

ImageMagick 画像ファイルの表示/処理を行うツール群.

xfig Brian Smith 氏らによるベクトル画像作成用プログラム.

tgif William Chia-Wei Cheng 氏による 2次元描画プログラム Tgif.

tgif2tex Tgif の図中の文字列を \TeX で処理する変換プログラム.

ghostscript 日本語 PostScript インタプリタ・レンダラ・プレビューア.

gv PostScript インタプリタである Ghostscript の拡張フロントエンド.

付録 B

参考資料

B.1 L^AT_EX と直接関係のない参考資料

L^AT_EX を使うには Unix 系 OS のツールや Unix 系 OS の基本的な操作ができる事が望ましいと思います。まずは Unix 系 OS の使い方に関する参考書です。

L^AT_EX を動かすための環境を構築するための技術を解説した資料です。Windows を手放せない方は Cgywin を、フリーにこだわる方は GNU Linux を、飽き足らない人は Unix の書籍を参照ください。

- [4] 佐藤竜一, いけだやすし, 野村直. Cygwin + CygwinJE-Windows で動かす UNIX——Cygwin is a UNIX environment for Windows. アスキー, 2003.
 - ▶ 日本語環境でも Cygwin がある程度動作するように追加がなされていますし, 大変参考になります.
- [5] 市川順一. デスクトップ Linux——Vine Linux 2.6 & OpenOffice.org. ローカス, 2004.
 - ▶ GNU Linux の中で日本語環境に配慮されたディストリビューションである Vine Linux の解説書です. 新しいです. 事務系ソフトの OpenOffice.org に関する情報もあるそうです.
- [6] Jerry Peek, Grace Todino, and John Strang. 入門 Unix オペレーティングシステム 第 5 版. オライリージャパン, 2002. 羽山博訳.
 - ▶ Unix システムの解説書です.
 シェル関連の書籍です. 本書で「端末」とか「ターミナル」と呼ばれるカーネルを突付くためのプログラムです. コンソールなどとも呼ばれます. Unix 系 OS ユーザは参照したほうが良いでしょう.
- [7] Cameron Newham and Bill Rosenblatt. 入門 bash 第 2 版. オライリージャパン, 1998. QUIPU LLC, 遠藤美代子訳.
 - ▶ bash に関する良書です. 中身はちょっと初心者向けではないかもしれませんが.
- [8] Paul DuBois. 入門 csh & tcsh. オライリージャパン, 2002. 鈴鹿倫之, 福澤康裕訳.
 - ▶ bash ではなく csh の本です.
- [9] Oliver Kiddle, Jerry Peek and Peter Stephenson. *From Bash to Z shell—Conquering the Command Line*—. Apress, 2004.
 - ▶ bash や tcsh の良いとこ取りをしたシェルです. 筆者はこのシェルを愛用しています.

大規模な文書を作成している方や複数人数で文書を作成している方にお勧めの書籍です。

- [10] Andrew Oram and Steve Talbott. *make* 改訂版. オライリージャパン, 1997. 矢吹道郎監訳. 菊池彰訳.
▶ 再コンパイル支援プログラム *make* の解説書です。これも初心者向けではないかもしれませんが。
- [11] Jennifer Vesperman. *実用 CVS*. オライリージャパン, 2003. 滝沢徹, 牧野祐子訳.
▶ 版管理プログラム *CVS* の解説書です。
- [12] Karl Fogel. *CVS——バージョン管理システム——*. オーム社, 2000.
▶ 版管理プログラム *CVS* の入門書です。これはウェブ上に日本語訳のマニュアルが公開されていると思います。
- コンピュータで文章を打ち込むためのプログラムをテキストエディッタと呼びます。特に Emacs であれば Windows (Meadow が開発されています) など多くの OS で動作します。
- [13] Debra Cameron, Bill Rosenblatt, and Eric Raymond. *入門 GNU Emacs 第 2 版*. オライリージャパン, 1999. 福崎俊博訳.
▶ テキストエディッタというか何でも屋の GNU Emacs の解説書です。GNU Emacs に関してウェブ上に日本語化されたマニュアルがあると思われま。ただし数 100 ページに及ぶと思うので、紙に印刷されたものもあると便利でしょう。
- [14] 小関吉則. *入門 Meadow/Emacs*. オーム社, 2003.
▶ GNU Emacs を Windows 上に移植した Mule の後継の Meadow の入門書です。Emacs の内容も含まれるそうです。Emacs と Meadow も基本的には同じ機能を有していると思われます。
- コンピュータがどのように文章を理解するのか、それが分かるとコンピュータに仕事をさせ易くなります。文字列処理やプログラミング言語を使うときさまざまな文章の加工ができるようになります。
- [15] Jeffrey E. F. Friedl. *詳説 正規表現 第 2 版*. オライリージャパン, 2003. 田和勝訳.
▶ 正規表現は多くのプログラミング言語で使われている言語表現形式です。正規表現は言語学に近い内容かもしれないです。
- [16] Dale Dougherty and Arnold Robbins. *sed & awk プログラミング 改訂版*. オライリージャパン, 1997. 福崎俊博訳.
▶ 古くから使われているテキスト加工プログラム、ストリームエディッタです。ほとんどの Unix 系 OS に導入されている標準的なプログラムです。
- [17] Randal L. Schwartz and Tom Phoenix. *初めての Perl 第 3 版*. オライリージャパン, 2003. 近藤嘉雪訳.
▶ スクリプト系プログラミング言語 Perl の入門書です。『プログラミング Perl』[18,19] の 2 冊よりは初心者向けの本です。Perl は CGI やテキスト処理など幅広い分野に応用されている優れた言語です。
- [18] Larry Wall, Tom Christiansen, and Jon Orwant. *プログラミング Perl 第 3 版 VOLUME 1*. オライリージャパン, 2002. 近藤嘉雪訳.
▶ Perl の開発者による公式なマニュアルの 1 巻目です。
- [19] ———. *プログラミング Perl 第 3 版 VOLUME 2*. オライリージャパン, 2002. 近藤嘉雪訳.
▶ Perl の開発者による公式なマニュアルの 2 巻目です。

- [20] 原信一郎. Ruby プログラミング入門. オーム社, まつもとゆきひろ監修.
▶ 国産のスクリプト系オブジェクト指向プログラミング言語 Ruby の入門書です. もちろん日本語の扱いが丁寧だし, Perl に取って代わるかもしれない言語です.
- [21] まつもとゆきひろ, 石塚圭樹. オブジェクト指向スクリプト言語 Ruby. アスキー, 1999.
▶ プログラミング初心者向けとは言い難い ruby の本です.
- [22] Ken Lunde. CJKV 日中韓越情報処理. オライリージャパン, 2002. 小松章 逆井克己訳.
▶ 『日本語情報処理』の改訂版として出版された非常に分厚い本です. CJKV 情報処理に関する数少ない良書だと思います.
ウェブに関する技術もあると最新の情報を即座に入手できるなどの利点があります. ウェブは情報の検索にも役立ちます.
- [23] Chuck Musciano and Bill Kennedy. HTML & XHTML 第 5 版. オライリージャパン, 2003. 原隆文訳.
▶ HTML を使いこなすというよりは現在の HTML の詳細な仕様書と言った感じです.
- [24] Tara Calishain and Rael Dornfest. Google Hacks——プロが使うテクニック & ツール 100 選. オライリージャパン, 2003. 山名早人監訳, 田中裕子訳.
▶ 検索エンジン Google の活用術を紹介した書籍です.
Adobe 社の開発した PostScript と PDF について知ると, ページ記述言語の特性や機能などが分かります.
- [25] Adobe Systems. PDF リファレンス第 2 版——Adobe Portable Document Format Version 1.3. ビアソンエディケーション, 2001.
▶ 数少ない PDF に関する日本語の技術資料です. Acrobat に関する書籍は沢山ありますが, PDF の規格そのものに言及したものはこれ以外にないと思われ.
- [26] ———. PostScript リファレンスマニュアル第 3 版——ASCII 電子出版シリーズ. アスキー, 2001. 桑沢清志訳.
▶ Adobe 社の開発したページ記述言語 PostScript の入門書です.
- [27] ———. ページ記述言語 PostScript プログラム・デザイン——電子出版シリーズ. アスキー, 1990. 松村邦仁 アスキー出版技術部訳.
▶ Adobe 社の開発したページ記述言語 PostScript の入門書の次に読むべき書籍です.
- [28] Sid Steward. PDF Hacks——文書作成、管理、活用のための達人テクニック. オライリージャパン, 2005. 千住 治郎訳.
▶ PDF の活用方法を解説した良書です. 訳書では日本語を通すための設定等も含まれています.
描画やプロットなどのグラフィックに関わる書籍です*1.
- [29] 川原稔. gnuplot パーフェクト・マニュアル. ソフトバンク・パブリッシング, 1999.
▶ プロットソフト Gnuplot の解説書です.
- [30] 皆本晃弥, 坂上貴之. GIMP/GNUPLOT/Tgif で学ぶグラフィック処理——UNIX グラフィックツール入門——. サイエンス者, 1999.

*1 MATLAB とか Octave なんかの情報も載せたほうが良いでしょうか？

▶ 実際には読んだ事がないので詳細は分かりませんが GIMP, Gnuplot, Tgif を取り扱った書籍です。

[31] 向井領治, 古川泰弘. GIMP エッセンシャルテクニック. オーム社, 2000.

▶ 描画プログラム GIMP の解説書です。

B.2 L^AT_EX の書籍

L^AT_EX に関する書籍は多く存在します。その中でも特に読むべきもの、手に入りやすいものを紹介します。

▼ B.2.1 入門書その 1

まずは入門として読むべき良書を紹介します。

[32] 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.

▶ 通称『奥村本』は定評があります。定期的に改訂がされているので、そのときの最新の情報も入手できます。

[33] 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. 阿瀬はる美訳。

▶ L^AT_EX の産みの親が執筆した良書。通称 *L^AT_EX manual* とも呼ばれている。上記の奥村晴彦氏の文献を購入していれば特に困る事もないのだが、L^AT_EX の基本をがっちり押さえた人向け。ただし、日本語環境の情報がないので別の情報が必要になると思います。

[34] Michel Goossens, Frank Mittelbach, and Alexander Samarin. The L^AT_EX コンパニオン. アスキー, 1998. アスキー書籍編集部監修。

▶ *L^AT_EX manual* [33] で解説できなかった事の補足説明と L^AT_EX 2_ε で使用できるマクロパッケージの紹介がされています。

[35] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. L^AT_EX グラフィックスコンパニオン——T_EX と PostScript による図形表現テクニック. アスキー, 2000. 鷺谷好輝訳。

▶ 『The L^AT_EX コンパニオン』[34] で解説できなかった事の補足説明と PostScript 周辺の技術を解説したものです。

[36] 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. 鷺谷好輝訳。

▶ PDF フォーマットの解説から HTML や XML と T_EX をどのように統合するかが解説されています。

[37] 白田昭司, 伊藤敏, 井上祥史. Linux 論文作成術. オーム社, 1999.

▶ Linux 環境を前提としていますが L^AT_EX や Gnuplot, Tgif などの解説を含んでいますので重宝すると思います。

[38] 生田誠三. L^AT_EX 2_ε 文典. 朝倉書店, 2000.

▶ L^AT_EX 2_ε ではこんな事もできるのかと感心してしまう 1 冊です。入力と出力が対になった辞典に近いと思います。

[39] 藤田眞作. L^AT_EX 2_ε コマンドブック. ソフトバンクパブリッシング, 2003.

▶ その名の通りコマンドブックです。例題が多すぎる気もしますが、その分学習しながら読み進める事もできます。

[40] 本田知亮. *L^AT_EX 2_ε 標準コマンド ポケットリファレンス*. 技術評論社, 2005.

▶ *L^AT_EX* の体裁調整に関して解説した良書です。他力本願的というよりは自作派向けのまとめ方になっています。

▼ B.2.2 入門書その 2

少し古くなったのですが、良書ですので紹介します。これらの書籍を購入しても最近の情報をインターネットなどで収集するのが良いと思います。

[41] 乙部厳己, 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.

▶ *pL^AT_EX 2_ε* の丁寧な解説書です。持っていて損のない本です。

[42] ———. *pL^AT_EX 2_ε for Windows Another Manual Vol.2 Extended Kit*. ソフトバンク, 1997.

▶ 上記で取り上げられなかった内容を紹介しています。Gnuplot や *AMSL^AT_EX* に関する情報もあります。

[43] 江口庄英. *Ghostscript Another Manual*. ソフトバンク, 1997.

▶ Ghostscript に関する数少ない解説書で、貴重な書籍です。

[44] 藤田眞作. *L^AT_EX 2_ε 階梯 第 2 版*. ピアソン・エデュケーション, 2000.

▶ *L^AT_EX 2_ε* の基本的な部分から *X^ML^AT_EX* やマクロの使用法、カウンタの使い方といったまで、洗練された一冊です。持っていて損のない一冊です。

[45] ———. *pL^AT_EX 2_ε 入門・縦横文書術*. ピアソン・エデュケーション, 2000.

▶ 上記の書籍を購入した後に縦組みにも興味のある人は購入すると良いでしょう。

[46] 中野賢. *日本語 L^AT_EX 2_ε ブック*. アスキー, 1996.

▶ 日本語 *L^AT_EX* の開発者による解説書です。これは良書なのですが、入手は難しいようです。

▼ B.2.3 数学系

入門の書籍を持っていれば困る事はないと思いますが、数式を多用する方は *AMSL^AT_EX* に関する書籍があったほうが便利だと思います。

[47] 嶋田隆司. *L^AT_EX 2_ε 数式環境*. シイエム・シイ出版部, 2001.

▶ *AMSL^AT_EX* に関する基本的な情報はこの 1 冊あれば良いでしょう。少々説明不足だと感じる部分があるのが残念ですが、論文作成には役に立つと思います。

[48] George Grätzer. *Math into L^AT_EX*. Springer, 2000.

▶ 英語ですがとても参考になる 1 冊です。CTAN: info/mil/ にサンプルがあります。

▼ B.2.4 化学・生化学

化学系では化学構造式や反応式などを書く必要があると思います。そのような図を作成するには藤田眞作氏が作成された $\text{X}\text{M}\text{T}_{\text{E}}\text{X}$ を使うのが良いと思います。

[49] 藤田眞作. $\text{X}\text{M}\text{T}_{\text{E}}\text{X}$: *typesetting chemical structural formulas*. 星雲社, 1997.

▶ 英語と日本語の 2ヶ国語で書かれた書籍で、海外でも広く使われていると思われます。バージョンアップされているので藤田眞作氏のホームページを確認したほうが良いでしょう。

[50] ———. 化学者・生化学者のための $\text{E}\text{T}_{\text{E}}\text{X}$ ——パソコンによる論文作成の手引き. 東京化学同人, 1993.

▶ 少し古いので今の事情に従わない部分もあるかもしれません。

▼ B.2.5 マクロやクラスの作成

$\text{E}\text{T}_{\text{E}}\text{X}$ をしばらく使っていると、その内部の機構について知りたくなるときがあるかもしれません。しかし `book.cls` などの中を見て溜め息をつきたくなる人も多いでしょう。そのようなときはマクロ作成やクラス作成を解説した書籍を参考にすると良いと思います。

[51] ページ・エンタープライゼス株式会社. $\text{E}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 【マクロ&クラス】プログラミング 基礎編. ページエンタープライゼス, 2002.

▶ かなりお勧めの書籍です。マクロ&クラス作成に関しては、この本があれば今まで悩んでいた事が解決すると思います。

[52] 吉永徹美. $\text{E}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 【マクロ&クラス】プログラミング 実践編. ページエンタープライゼス, 2003.

▶ 上記の続編です。

▼ B.2.6 $\text{T}_{\text{E}}\text{X}$ についての本

$\text{E}\text{T}_{\text{E}}\text{X}$ を追求しているとやはり $\text{T}_{\text{E}}\text{X}$ についても知りたくなります。Donald Knuth や plain $\text{T}_{\text{E}}\text{X}$ に関する書籍を参考にすると良いでしょう。いずれの本も入手が難しくなっていますので、早めに購入するか古本屋で探してください。

[53] Donald Knuth. 改訂新版 $\text{T}_{\text{E}}\text{X}$ ブック. アスキー, 1992. 斎藤信男監修. 鷺谷好輝訳。

▶ $\text{T}_{\text{E}}\text{X}$ の作者が書いた本です。アスキーでは取り扱いを停止している模様です。非常に重要な 1冊です。

[54] ———. $\text{M}\text{E}\text{T}\text{A}\text{F}\text{O}\text{N}\text{T}$ ブック. アスキー, 1994. 鷺谷好輝訳。

▶ $\text{T}_{\text{E}}\text{X}$ システムに必要なだったフォントの作成をするためのプログラムの解説です。これも重要な 1冊です。

B.3 文書作成全般

▼ B.3.1 作文技術

作文技術や校正に関する知識もあれば執筆・編集作業の効率が向上する事があります。

- [55] 木下是雄. 理科系の作文技術. 中公新書 624. 中央公論社, 1981.
- [56] 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.
- [57] 小笠原喜康. 大学生のためのレポート・論文術. 講談社, 2002.
- [58] 大隈秀夫. 分かりやすい日本語の書き方. 講談社, 2004.

▼ B.3.2 組版全般

組版に関する知識がなければまともな本は作れませんので、以下の書籍を読んでみる事をお勧めします。

- [59] 日本エディタースクール編集. 文字の組方ルールブック——横組編. 日本エディタースクール出版部, 2001.
 - ▶ 横組における組版規則を要領よくまとめた手軽な良書です.
- [60] ——. 文字の組方ルールブック——縦組編. 日本エディタースクール出版部, 2001.
 - ▶ 縦組における組版規則を要領よくまとめた手軽な良書です.
- [61] ——. 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出版部, 1999.
 - ▶ 校正記号の使い方をコンパクトにまとめた良書です. 以上の3冊は値段も手ごろです.
- [62] ——. 新編 出版編集技術 上巻. 日本エディタースクール出版部, 1997.
 - ▶ 内容が少し古いのですが(写植時代の名残が強いので)実技的な部分も多く含んだ出版編集に関わる人間は持っていて損のない1冊と言えます. これはその上巻です.
- [63] ——. 新編 出版編集技術 下巻. 日本エディタースクール出版部, 1997.
 - ▶ これは上記の下巻です.
- [64] Robert Ritter. *The Oxford Style Manual*. Oxford University Press, 2002.
 - ▶ 欧文組版に関する優れた資料です. 組版関係の事に興味があれば読んでみてください. こちらは英語圏での標準の組版規則についての解説です.
- [65] University of Chicago Press Staff. *The Chicago Manual of Style 15th edition*. Chicago University Press, 2003.
 - ▶ こちらは米語圏での標準の組版規則についての解説です.

▼ B.3.3 多少入手が難しい書籍

10年も経つと入手が難しくなるのが専門書というもので、売っているときに買わなければ2度と手に入らないかもしれません。

- [66] 藤田眞作. \LaTeX まくろの八衢. アジソン・ウェスレイ, 1995.
 ▶ 繰り返し処理やカウンタの使い方などを習得するために是非読んでおきたい本です.
- [67] ———. \LaTeX 本づくりの八衢. アジソン・ウェスレイ, 1996.
 ▶ 和文組版に配慮した本づくりに関した技術を取り扱った本です.

▼ B.3.4 無料の冊子

とにかく無料で済ませたい方はオンラインで公開されている無料の冊子をご自分で印刷して勉強されるのが良いでしょう.

- [68] Tobias Oetiker (Hubert Partl, Irene Hyna and Elisabeth Schlegl) . *The Not So Short Introduction to \LaTeX 2 ϵ —Or \LaTeX 2 ϵ in 129 minutes*.
 ▶ CTAN: [info/lshort/english/](http://www.ctan.org/info/lshort/english/)
 ▶ 通称 lshort と呼ばれる \LaTeX の入門書です. これだけ読めば一通り \LaTeX が使えるようになると思います.
- [69] Tobias Oetiker. \LaTeX 2 ϵ への道——83 分 \LaTeX 2 ϵ 入門. 野村昌孝訳.
 ▶ CTAN: [info/lshort/japanese/](http://www.ctan.org/info/lshort/japanese/)
 ▶ 上記の lshort の日本語訳で通称 jlshort と呼ばれています.
- [70] 岩熊哲夫, 古川徹生. \LaTeX のマクロやスタイルファイルの利用. 1994.
 ▶ <http://www.civil.tohoku.ac.jp/~bear/node12.html>
 ▶ \LaTeX の旧版 \LaTeX 2.09 でのマクロの利用についてです. 今でも十分参考になると思います.

B.4 ウェブの資料

近年はウェブ (World Wide Web) というネットワークが広く活用されています. 情報の共有や多様なメディアの共存などができるウェブ上では実に様々な情報が提供されています. しかし, あまりに沢山散在するために, どこをどう探せば良いのかが分かりづらいのも事実です. そのような場合は検索エンジンと呼ばれる無料の検索機構やディレクトリと呼ばれる検索対象を項目ごとに階層的に分類したウェブページなども存在します. これらの中で著者のおすすめは Google^{*2}です.

▼ B.4.1 CTAN と Ring Server の使い方

CTAN とは Comprehensive \TeX Archive Network の略で \TeX に関連するマクロパッケージやプログラムなどを収集しそれを提供するサイトです. 英語版ですが CTAN には Graham Williams によるカタログ (CTAN: [help/Catalogue/index.html](http://www.ctan.org/help/Catalogue/index.html)) がありますので, そこから自分がしたい事を実現できるパッケージなどを見つけると良いでしょう.

Ring Server は (ネットワーク) 社会にとって有用だと思われるソフトウェアとその開発を支援

^{*2} <http://www.google.co.jp>

するプロジェクトが運営するファイルサーバ群です。 <http://www.dnsbalance.ring.gr.jp/> にアクセスすると自動的に空いているサーバに接続できます。アクセスすると Ring Server のトップページを閲覧する事ができるはずです。「ソフトウェアライブラリ」や「検索」等からお目当てのファイルを探す事ができます。直接 <http://www.ring.gr.jp/pub/> にアクセスすれば、CPAN/, GNU/, linux/, text/ 等のディレクトリ*3があります。特に有益だと思われるディレクトリを抜粋したものを図 B.1 に示します。

▼ B.4.2 L^AT_EX

L^AT_EX や L^AT_EX の基本マクロ、L^AT_EX の拡張マクロの情報源です。

- [71] Johannes Braams, David Carlisle, Alan Jeffrey, Leslie Lamport, Frank Mittelbach, Chris Rowley, and Rainer Schöpf. *L^AT_EX 2_ε Sources*, 2001.
 - ▶ CTAN: `macros/latex2e/base/source2e.tex`
 - ▶ L^AT_EX 2_ε の全ソースコードです。大変勉強になります。
- [72] David Carlisle. *The enumerate package*, 1999.
 - ▶ CTAN: `macros/latex2e/required/tools/enumerate.dtx`
 - ▶ L^AT_EX tools に含まれるパッケージです。番号付きリスト環境の拡張です。
- [73] ———. *The longtable package*, 2000.
 - ▶ CTAN: `macros/latex2e/required/tools/longtable.dtx`
 - ▶ L^AT_EX tools に含まれるパッケージです。ページをまたぐ程の大きな表を作成するために使います。
- [74] ———. *The ifthen package*, 2001.
 - ▶ CTAN: `macros/latex2e/base/ifthen.dtx`
 - ▶ L^AT_EX tools に含まれるパッケージです。条件分岐などに使えます。
- [75] Kent McPherson. *Displaying page layout variables*, 2000.
 - ▶ CTAN: `macros/latex2e/required/tools/layout.dtx`
 - ▶ 現在使用中のクラスでのページレイアウトについての情報を出力するためのマクロです。
- [76] Frank Mittelbach. *Producing slides with L^AT_EX 2_ε*, 1997.
 - ▶ CTAN: `macros/latex2e/base/slides.dtx`
 - ▶ slides クラスについての情報です。
- [77] ———. *An environment for multicolumn output*, 2003.
 - ▶ CTAN: `macros/latex2e/required/tools/multicol.dtx`
 - ▶ 多段組を実現する multicol パッケージについてです。
- [78] ———. *An Extension of the L^AT_EX theorem environment*, 2003.
 - ▶ `macros/latex2e/required/tools/theorem.dtx`
 - ▶ L^AT_EX の theorem 環境を拡張した theorem パッケージです。A_MS_LA_TE_X のものよりも高性能かもしれません。

*3 <http://www.ring.gr.jp/pub/text/>

- CTAN/ CTAN: *Comprehensive T_EX Archive Network*. ここから探検が始まります.
 - documentation/ T_EX に関連する文書が蓄積されています.
 - * epslatex.pdf Keith Reckdahl 氏による L^AT_EX での画像の張り込みに関する解説 *Using Imported Graphics in L^AT_EX and pdfL^AT_EX*.
 - * gentle/ Michael Doob 氏による *A Gentle Introduction to T_EX—A Manual for Self-study*. L^AT_EX ではなく plain T_EX についての解説書です.
 - * impatient/ Paul W. Abrahams 氏らによる plain T_EX の解説書 *T_EX for the Impatient*. 本書と同じ *The GNU Free Documentation License* のマニュアル. 旧版で邦訳 (ISBN: 479529643X) も出版されています.
 - * lshort/ Tobias Oetiker 氏らによる L^AT_EX 2_ε の入門書 *The Not So Short Introduction to L^AT_EX 2_ε*.
 - japanese/ 野村昌孝氏による日本語訳があります.
 - * mil/ George Grätzer 氏による数式について特記した入門書 *Math into L^AT_EX—An Introduction to L^AT_EX and A_MS_LT_EX*. 第 3 版 (ISBN: 0817641319) が出版されています.
 - * symbols/ L^AT_EX で使える記号類の一覧が閲覧できます.
 - comprehensive/ Scott Pakin 氏による L^AT_EX で使用できる記号の一覧 *The Comprehensive L^AT_EX Symbol List*.
 - dviware/ デバイスドライバ等があります.
 - * dvipdfmx/ 平田俊作氏と趙珍煥氏による Dvipdfm の拡張 Dvipdfmx です.
 - fonts/ T_EX に関するフォント等があります.
 - * jknappen/ Jörg Knappen 氏による *European Computer Modern Fonts* (通称 EC フォント) があります.
 - macros/ T_EX で使用できる便利なマクロパッケージがここに収録されています.
 - * latex/ L^AT_EX 2_ε で使用できるマクロパッケージがあります. 現在は latex2e というディレクトリですが, L^AT_EX の旧版のマクロは latex209 というディレクトリに, 次期 L^AT_EX 3 の場合は latex3 なるディレクトリに蓄積されるだろうと思います.
 - base/ L^AT_EX 2_ε の基本となるファイル群があります.
 - contrib/ 世界中の T_EX ユーザから投稿されたマクロパッケージがあります.
 - doc/ L^AT_EX 3 プロジェクトチームから公式に配布される文書です.
 - required/ A_MS_LT_EX, babel, graphicx 等の重要なマクロです.
 - support/ 何らかの形で役立つツール等があります.
 - * latexmk/ David J. Musliner 氏による L^AT_EX の再コンパイル支援プログラムです.
 - * pdfcrop/ Heiko Oberdiek 氏による PDF の余白を切り抜く Perl スクリプト (要 pdfL^AT_EX, Perl, Ghostscript) です.
 - systems/ 環境 (オペレーティングシステム) に依存するファイル群です.
 - * win32/ Windows 環境に依存するプログラム等です.
 - tds/ TUG: *T_EX Users Group* による TDS に関する文書です.
- TeX/ CTAN とは別に, Ring Server が収録している T_EX 関連のものです.
 - ascii-ptex/ アスキーによる日本語化された pT_EX/pL^AT_EX です.
 - dviout/ 大島利雄氏による Windows 用 DVI プレビューア dviout です.
 - ptex-win32/ 角藤亮氏による Windows 用の pT_EX とその周辺ツールです.
 - * current/ 角藤版 pT_EX の最新版です.
 - * gs/ 日本語化済みの Ghostscript 等があります.

図 B.1 Ring Server の探検

- [79] Frank Mittelbach, Denys Duchier, Johannes Braams, Marcin Wolinski, and Mark Wooding. *The docStrip program*, 1999.
- ▶ CTAN: `CTAN/macros/latex2e/base/docstrip.dtx`
 - ▶ クラス作成者は必読の DocStrip ユーティリティについての解説です。
- [80] Rainer Schöpf, Bernd Raichle and Chris Rowley. *A New Implementation of L^AT_EX's verbatim and verbatim* Environments*, 2001.
- ▶ CTAN: `macros/latex2e/required/tools/verbatim.dtx`
 - ▶ `verbatim` 環境の拡張についてです。
- [81] L^AT_EX 3 Project Team. *L^AT_EX 2_ε font selection*, 2000.
- ▶ CTAN: `macros/latex2e/base/fntguide.tex`
 - ▶ L^AT_EX でのフォントの選択方法 (NFSS) についての文書です。L^AT_EX で使われているフォントの定義の仕方について知りたいならば読むべきでしょう。
- [82] ———. *L^AT_EX 2_ε for authors*, 2001.
- ▶ CTAN: `macros/latex2e/base/usrguide.tex`
 - ▶ L^AT_EX を使い始める人は読むべき情報です。L^AT_EX の更新履歴などが含まれているので重要な文書です。
- [83] ———. *L^AT_EX 2_ε for class and package writers*, 1999.
- ▶ CTAN: `macros/latex2e/base/clsguide.tex`
 - ▶ クラスファイルやマクロパッケージを設計する人向けの解説です。
- [84] Kresten Thorup, Frank Jensen, and Chris Rowley. *The calc package*, 1998.
- ▶ CTAN: `macros/latex2e/required/tools/calc.dtx`
 - ▶ L^AT_EX での計算を楽にするパッケージです。
- [85] Vlandimir Volovich, Werner Lemberg and L^AT_EX 3 Project Team. *Cyrillic languages support in L^AT_EX*, 1999.
- ▶ CTAN: `macros/latex2e/base/cyrguide.tex`
 - ▶ キリル文字 (ロシア語) を扱うための解説です。

▼ B.4.3 L^AT_EX 周辺の資料

L^AT_EX 周辺の技術資料です。

- [86] Mark Wicks. *Dvipdfm User's Manual*, 1999.
- ▶ CTAN: `dviware/dvipdfm/dvipdfm.pdf`
 - ▶ デバイスドライバ Dvipdfm のマニュアルです。
- [87] 趙珍煥. *DVIPDFMx, an eXtension of DVIPDFM*, 2003.
- ▶ <http://project.ktug.or.kr/dvipdfmx/>
 - ▶ Dvipdfm の拡張版である Dvipdfmx についての情報です。
- [88] Oren Patashnik. BibL^AT_EXing: BibL^AT_EX の使い方, 1991. 松井正一訳。
- ▶ jBibL^AT_EX と共に配布される文書です。
- [89] 松井正一. 日本語 BibL^AT_EX: jBibL^AT_EX, 1991.
- ▶ jBibL^AT_EX と共に配布される文書です。

- [90] Scott Pakin. *How to Package Your L^AT_EX Package*, 2003.
- ▶ CTAN: [info/dtxut/](http://ctan.org/info/dtxut/)
 - ▶ 自分が作成した L^AT_EX のパッケージを配布するための、パッケージの作成方法が書かれた解説です。
- [91] Philipp Lehman. *The Font Installation Guide*, 2003.
- ▶ CTAN: [info/Type1fonts/fontinstallationguide/](http://ctan.org/info/Type1fonts/fontinstallationguide/)
 - ▶ L^AT_EX で PostScript フォントを使うための解説です。
- [92] Karl Berry. *Fontname*, 2003.
- ▶ CTAN: [info/fontname/](http://ctan.org/info/fontname/)
 - ▶ L^AT_EX におけるフォント名についての解説です。
- [93] Walter Schmidt. *Using common PostScript fonts with L^AT_EX*, 2004.
- ▶ CTAN: [macros/latex/required/psnfss/psnfss2e.pdf](http://ctan.org/macros/latex/required/psnfss/psnfss2e.pdf)
 - ▶ L^AT_EX で PostScript フォントを使うための解説資料です。
- [94] T_EX Users Group. *A Directory Structure for T_EX Files*, 2003.
- ▶ CTAN: [tds/](http://ctan.org/tds/)
 - ▶ T_EX に関連するファイルが乱雑に分類されていたので、ある基準できちんと整理するように定めた資料です。

▼ B.4.4 マクロパッケージ

- [95] Carsten Heinz. *The Listings Package*, 2003.
- ▶ CTAN: [macros/latex2e/contrib/listings/](http://ctan.org/macros/latex2e/contrib/listings/)
 - ▶ ソースコードを整形する listings パッケージについてです。
- [96] Sebastian Rahtz. *Hypertext marks in L^AT_EX the hyperref package*, 1998.
- ▶ CTAN: [macros/latex2e/contrib/hyperref/](http://ctan.org/macros/latex2e/contrib/hyperref/)
 - ▶ L^AT_EX でハイパーリンクを実現するためのマクロ hyperref に関しての資料です。
- [97] Keith Reckdahl. *Using Imported Graphics in L^AT_EX and pdfL^AT_EX*, 2006.
- ▶ CTAN: [info/epslatex.pdf](http://ctan.org/info/epslatex.pdf)
 - ▶ 画像を取り込むための graphicx (graphics) パッケージの使い方を丁寧に解説した文書です。
- [98] Timothy Zandt. *PSTricks: PostScript macros for Generic T_EX*, 1993.
- ▶ <http://tug.org/PSTricks/main.cgi/>
 - ▶ PostScript 命令を使って図形を描く pstricks パッケージについての文書です。
- [99] Piet Oostrum. *Page layout in L^AT_EX*.
- ▶ CTAN: [macros/latex2e/contrib/fancyhdr/](http://ctan.org/macros/latex2e/contrib/fancyhdr/)
 - ▶ ヘッダーやフッターを調整する fancyhdr についてのマニュアルです。
- [100] American Mathematical Society. *User's Guide for amsmath Package*.
- ▶ <http://www.ams.org/tex/amslatex.html>
 - ▶ 米国数学会が提供する A_MS_LT_EX に含まれる amsmath パッケージに関する解説です。

B.5 ウェブページ

インターネットは広大で L^AT_EX に関する情報がウェブページで沢山公開されています。それらの情報を追いかけるのも良いでしょう。

[URL-A] T_EX Wiki

- ▶ <http://oku.edu.mie-u.ac.jp/~okumura/texwiki/>
- ▶ 奥村晴彦氏が管理しているページです。

[URL-B] The Publishing T_EX

- ▶ <http://www.ascii.co.jp/pb/ptex/>
- ▶ 日本語 T_EX を開発したアスキー社のウェブページです。日本語 T_EX についての情報があります。

[URL-C] W32T_EX

- ▶ <http://www.fsci.fuk.kindai.ac.jp/~kakuto/win32-ptex/>
- ▶ 角藤亮氏が Windows 用に移植された pT_EX をダウンロードできます。

[URL-D] dviout/dviprt 情報

- ▶ <http://akagi.ms.u-tokyo.ac.jp/index-j.html>
- ▶ Windows 用の DVI プレビューア dviout を開発している大島利雄氏のホームページ。

[URL-E] Mac pT_EX とその周辺

- ▶ <http://macptex.appi.keio.ac.jp/~uchiyama/macptex.html>
- ▶ 内山孝憲氏が管理されている Mac OS X 上で動作する Mac pT_EX を取り扱ったホームページ。

[URL-F] pL^AT_EX for Windows

- ▶ <http://www17.plala.or.jp/ohishi-masaru/>
- ▶ 大石勝氏のホームページ。L^AT_EX の周辺の情報も豊富です。

[URL-G] 藤田眞作 個人ページ

- ▶ <http://imt.chem.kit.ac.jp/fujita/fujitas/fujita.html>
- ▶ 藤田眞作氏のホームページ。いわゆる『藤田本』と呼ばれる良書の著者です。X_YL^AT_EX [49] の開発もしています。

[URL-H] おとべよしき 乙部厳己個人ページ

- ▶ <http://argent.shinshu-u.ac.jp/otobe/>
- ▶ 乙部厳己氏のホームページ。いわゆる『乙部本』と呼ばれる良書の著者です。

[URL-I] yama-Ga.com

- ▶ <http://www.yama-ga.com/>
- ▶ 山賀正人氏のホームページ。Gnuplot の日本語化など。

[URL-J] 野鳥 (YaT_EX)

- ▶ <http://www.yatex.org/>
- ▶ やちよう 野鳥は広瀬雄二氏が開発した GNU Emacs 上で使える elisp です。

[URL-K] T_EX 「超」 入門

- ▶ <http://www.nsknet.or.jp/~tony/TeX/texindex.html>
- ▶ 刀祢宏三郎氏のホームページ。

[URL-L] 初等数学プリント作成マクロ emath

- ▶ <http://emath.s40.xrea.com/>
- ▶ 数学のプリントを作成するだけでなく、数学の多方面にも活用できそうなマクロを収めた emath パッケージの配布元です。

[URL-M] ワープロユーザーのための L^AT_EX 入門

- ▶ <http://www.klavis.info/texindex.html>
- ▶ 大友康寛氏が開設している初心者向けの L^AT_EX の情報。インストール等は丁寧に画像付きですから詳しいです。

[URL-N] SMALL L^AT_EX LAB

- ▶ <http://www.h4.dion.ne.jp/~latexcat/>
- ▶ マクロ作成の良書 [51, 52] を手がけられた吉永徹美氏のホームページです。

[URL-O] L^AT_EX 2_ε 的

- ▶ <http://psitau.at.infoseek.co.jp/>
- ▶ さまざまなマクロを公開している齋藤修三郎氏のホームページです。特に L^AT_EX でユニコード文字や OpenType フォントを使う事ができる utf/otf パッケージはとても便利です。

[URL-P] 熊澤吉紀のホームページ

- ▶ <http://www.biwako.shiga-u.ac.jp/sensei/kumazawa/>
- ▶ ソースと画像でマクロの使用例がある熊澤吉紀氏のホームページです。

[URL-Q] 竹野研究室 Home Page

- ▶ <http://takeno.iee.niit.ac.jp/~foo/>
- ▶ latex2html の日本語化や Gnuplot のマニュアルの日本語化などをされている竹野茂治氏のホームページです。

[URL-R] Ghostscript 8.53 + GSview 4.8 の日本語版

- ▶ <http://auemath.aichi-edu.ac.jp/~khotta/ghost/>
- ▶ 題名にとらわれずに pL^AT_EX 2_ε についての情報を提供している堀田耕作氏のホームページです。

[URL-S] L^AT_EX によるドイツ語・日本語処理

- ▶ <http://www.lg.fukuoka-u.ac.jp/~ynagata/latex.html>
- ▶ 福岡大学の永田善久氏が管理されているホームページです。本書では多言語処理についてはほとんど扱っておりません。申し訳ないのですがウェブからそれらの情報を集めてみてください。

[URL-T] 日本語 L^AT_EX による多言語処理

- ▶ <http://www2.tba.t-com.ne.jp/ing/>
- ▶ 上記の永田氏のページに比べてこちらの稲垣徹氏のホームページでは日本語 L^AT_EX 環境での多言語処理を念頭に置かれた解説があります。

付録 C

変更履歴

この文書は私一人で執筆しておりますから、どこかに間違いや誤植がある確率が高くなっています。「あれっおかしいな？」と思う箇所がありましたら私のホームページ^{*1}の掲示板かメールアドレス^{*2}にご連絡ください。

1.14 2016/07/23

大石勝様の URL を訂正。

1.13 2006/08/18

誤植訂正版。

1.12 2006/05/12

図 B.1 における敬称が省略されていたのを修正しました。

倍角ダーンの後の余計な空白を取り除きました。

B.4.4 節の fancyhdr パッケージのアドレスを修正しました。

1.11 2006/05/12

ver. 1.10 をさらに校正しました。

前付けにある『フリーソフトウェアとフリーマニュアル』を削除しました。これに伴い『謝辞』の直後のページに代わりとなる FSF とこの文書の位置付け及び PDF 版の本書の所在に関する説明を追加しました。

文書全体において語句・語調の統一を行いました。

1.10 2006/05/07

ver. 1.00 の誤植訂正版として配布しました。

1.00 2006/04/20

大幅な改訂を行い、方向性を若干「理工系の学生・研究者向け」としました。

ページレイアウトと使用しているマクロの改変を行いました。

B.4.1 節を追加しました。

付録 B の人名を包括的に索引に追加しました。

口絵を削除しました。

前付けにある『まえがき』の『凡例』を加筆し、『FUNNIST について少し』を削除しま

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

*2 thor@tex.dante.jp

した。

前付けにある『フリーソフトウェアとフリーマニュアル』の『Free Software Foundationとその活動について』を加筆しました。

章構成を入れ替えました。

3.6.3 節を修正しました。

第 5 章に例題を加筆しました。

5.11 節に \LaTeX に関する情報を追加しました。

6.4 節を加筆・修正しました。

8.20 節を修正しました。

付録 A を追加しました。

0.34 2005/03/20

0.33 の誤植訂正版という形で配布しました。

0.33 2004/12/28

誤植の訂正を行いました。

ライセンス的に *free* とは言いがたい画像を削除しました。

0.32 2004/01/14

誤植の訂正を行いました。

著者の連絡先が変更になったため、URL と e-mail を変更しました。

URL の変更にもない付録 B を修正しました。

0.31 2004/08/19

誤植の訂正を行いました。

改行が変な部分や索引の倍角ダッシュを修正しました。

URL の変更等にもない付録 B を修正しました。

0.30 2004/08/05

初級編に必要なと思われる部分を記述し、これ以上は修正しないという完成版に近いものを公開しました。

0.21 2004/04/30

誤植の訂正を行いました。

0.20 2004/04/16

誤植の訂正を行いました。

句読点を全角のピリオド・コンマに統一しました。

爪掛けについては章見出しも出力するようにしました。

第 5 章で空きに関する記述を加筆しました。

索引について抜けていた人名や語句の補充をしました。

0.10 2004/04/02

初版を発行しました。

付録 D

The 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

D.1 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 man-

uals; 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.

D.2 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 por-

tion 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 section 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 five 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, L^AT_EX 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 for-

ats 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.

D.3 Verbatim copying

You may copy and distribute the document in any medium, either commercially or non-commercially, 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 D.4.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

D.4 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 mate-

rial 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.

D.5 Modifications

You may copy and distribute a modified version of the doc-

ument under the conditions of sections D.3 and D.4 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.
- 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.

D.6 Combining documents

You may combine the document with other documents released under this license, under the terms defined in section D.5 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.’

D.7 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 docu-

ment.

D.8 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 D.4 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.

D.9 Translation

Translation is considered a kind of modification, so you may distribute translations of the document under the terms of section D.5. 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 D.5) to preserve its title (section D.2) will typically require changing the actual title.

D.10 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.

D.11 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*¹.

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.

*¹ <http://www.gnu.org/copyleft/>



命令索引

数字/記号

\surd 69, 120, 274
 $\!$ 120, 275
 $\" (i)$ 61
 $\#$ 12
 $\#$ 11
 $\$$ 12
 $\$$ 11, 15, 99, 114
 $\%$ 12
 $\%$ 11, 61, 99
 $\&$ 12
 $\&$ 11, 99
 array 環境の—— ... 126
 eqnarray*環境の—— ...
 116
 tabular 環境の—— ... 174
 $\prime (é)$ 61
 \langle 311
 \langle 114
 \rangle 311
 \rangle 114
 \setminus 119, 120, 275
 \sim 67
 $\dot{\ } (á)$ 61
 \surd 60
 \setminus 119, 120, 275
 \setminus 119, 120, 275
 $\bar{=}$ (\bar{a}) 61
 $\@$ 69
 $\@biblabel$ 324
 $\@empty$ 253
 $\@evenfoot$ 253
 $\@evenhead$ 253
 $\@gobbletwo$ 253
 $\@ifnextchar$ 131
 $\@ifstar$ 99
 $\@makecaption$ 324, 325
 $\@makechapterhead$ 324
 $\@makeschapterhead$ 324
 $\@mkboth$ 253
 $\@oddfont$ 253
 $\@oddhead$ 253

$\@startsection$ 324
 \lbracket 114
 \backslash 11, 91, 99
 \rbracket 114
 $\hat{\ } (ô)$ 61
 $\hat{\ }$ 11, 99, 121
 \sim 12
 $_$ 11, 99, 121
 $\grave{\ } (à)$ 61
 $\{$ 12, 125
 $\{$ 11, 99
 $\}$ 12, 125
 $\}$ 11, 99
 \sim 125
 $\tilde{\ } (\tilde{n})$ 61
 \sim 11, 70, 99, 275

A

\AA (\AA) 61
 \aa (\aa) 61
 abstract 環境 52
 $\acute{\ } (á)$ 135
 \Acute 161
 \addcontentsline 259
 \addtocontents 259
 \addtocounter 108, 307
 \addtolength 263, 307
 \addvspace 276
 \advance 307
 \AE (\AE) 61
 \ae (\ae) 61
 \AffixLabels 201, 202
 \afterpage 87
 \aleph (\aleph) 136
 alignat 環境 153, 162
 alignedat 環境 162
 aligned 環境 162
 align 環境 ... 153, 155, 162
 \allinethickness 209
 \allowdisplaybreaks .. 163
 alltt 環境 63
 \Alph 74, 108

\alph 74, 108
 α (α) 132
 αup (α) 169
 \amalg (\amalg) 134
 \and 47, 176, 311
 \angle (\angle) 136, 168
 \appendix 278
 \appendixname 259
 appendix 環境 278
 \approx (\approx) 134
 \approxeq (\approx) 166
 \arabic 74, 108
 \arc 210
 \arccos (\arccos) 122
 \arcsin (\arcsin) 122
 \arctan (\arctan) 122
 \arg (\arg) 122
 array 環境 87, 126, 153,
 173, 180, 184

\Arrowvert (\parallel) 125
 \arrowvert (\parallel) 125
 \ast ($*$) 134
 \asymp (\asymp) 134
 \AtBeginDvi 30, 38
 \AtEndDocument 253
 \atop 145
 \author 45, 46

B

\b (\b) 61
 \B 202
 \backepsilon (ϵ) 166
 \backprime (\prime) 168
 \backsim (\sim) 166
 \backsimeq (\simeq) 166
 \backslash (\backslash) ... 125, 136
 \bar{a} 135
 \Bar 161
 \barwedge ($\bar{\wedge}$) 165
 \baselineskip 243, 321
 \baselinestretch 55
 \Bbbk (\mathbb{k}) 168

- `Bcenter` 環境 271
`Bdescription` 環境 272
`\because` (∵) 166
`\begin` 20
`\begingroup` 325
`Benumerate` 環境 272
`Beqnarray` 環境 271
`\beta` (β) 132
`\betaup` (β) 169
`\beth` (\beth) 168
`\between` ($\text{\scriptsize between}$) 166
`Bflushleft` 環境 271
`Bflushright` 環境 271
`\bfseries` 80, 322
`\bgroup` 131
`\bibitem` 220, 221
`\bibliography` 224, 228
`\bibliographystyle` ... 224, 228
`\bibname` 259
`\Big` 126
`\big` 126
`\bigcap` (\cap) 134
`\bigcirc` (\bigcirc) 134
`\bigcup` (\cup) 134
`\bigl` 126
`\bigrm` 126
`\bigmplus` (\scriptsize mplus) 169
`\bigodot` (\odot) 134
`\bigoplus` (\oplus) 134
`\bigotimes` (\otimes) 134
`\bigr` 126
`\bigskip` 276
`\bigsqcap` (\sqcap) 169
`\bigsqcapplus` (\scriptsize plus) 169
`\bigsqcup` (\sqcup) 134
`\bigsqcupplus` (\scriptsize plus) 169
`\bigstar` (\star) 168
`\bigtriangledown` (∇) 134
`\bigtriangleup` (\triangle) 134
`\biguplus` (\scriptsize uplus) 134
`\bigvee` (\vee) 134
`\bigwedge` (\wedge) 134
`\binom` 164
`Bitemize` 環境 272
`\blacklozenge` (\blacklozenge) 168
`\blacksquare` (\blacksquare) 168
`\blacktriangle` (\blacktriangle) 168
`\blacktriangledown` (\blacktriangledown) 168
`\blacktriangleleft` (\blacktriangleleft) 166
`\blacktriangleright` (\blacktriangleright) 166
`\bm` 142
`Bmatrix` 環境 156
`bmatrix` 環境 156
`\bmod` 122
`\boldmath` 142
`\boldsymbol` 142, 150
`\boolean` 310
`\bordermatrix` 128
`\Bot` (\perp) 169
`\bot` (\perp) 136
`\bottomrule` 178
`\bowtie` (\bowtie) 134
`\Box` (\square) 137
`\boxast` (\boxast) 168
`\boxbar` (\boxbar) 168
`\boxbslash` (\boxbslash) 168
`\boxdot` (\boxdot) 165
`\boxdotLeft` (\boxdotLeft) 170
`\boxdotleft` (\boxdotleft) 170
`\boxdotRight` (\boxdotRight) 170
`\boxdotright` (\boxdotright) 170
`\boxLeft` (\boxLeft) 170
`\boxleft` (\boxleft) 170
`\boxminus` (\boxminus) 165
`\boxplus` (\boxplus) 165
`\boxput` 273
`\boxput*` 273
`\boxRight` (\boxRight) 170
`\boxright` (\boxright) 170
`\boxslash` (\boxslash) 168
`\boxtimes` (\boxtimes) 165
`\brace` 145
`\bracevert` (\bracevert) 125
`\brack` 145
`\breve` (\breve) 135
`\Breve` 161
`\bullet` (\bullet) 134
`\Bumpeq` (\Bumpeq) 166
`\bumpeq` (\bumpeq) 166
- ## C
- `\c` (\c) 61
`\Cap` (\Cap) 165
`\cap` (\cap) 134
`\caption` 109, 173
`\cases` 145
`cases` 環境 145, 153
`\cbezier` 211
`\cdot` (\cdot) 134
`\cdots` (\cdots) 136
`CD` 環境 150
`cemph` 環境 94
`\centerdot` (\centerdot) 165
`\centering` 72, 103, 324
`center` 環境 71, 72
`\cfoot` 251
`\cfrac` 164
`\chapter` ... 48, 49, 109, 318
`\chapter*` 52, 257
`\chaptermark` 252
`\chead` 251
`\check` (\check) 135
`\Check` 161
`\checkmark` (\checkmark) 168
`\chi` (χ) 132
`\chiup` (χ) 169
`\choose` 145
`\circ` (\circ) 134
`\circeq` (\circeq) 166
`\circle*` 206, 210
`\circlearrowleft` (\circlearrowleft) 167
`\circlearrowright` (\circlearrowright) 167
`\circledast` (\circledast) 165
`\circledbar` (\circledbar) 168
`\circledbslash` (\circledbslash) 168
`\circledcirc` (\circledcirc) 165
`\circleddash` (\circleddash) 165
`\circleddotleft` (\circleddotleft) 170
`\circleddotright` (\circleddotright) 170
`\circledgtr` (\circledgtr) 170
`\circledless` (\circledless) 170
`\circledR` (\circledR) 168
`\circledS` (\circledS) 168
`\circledvee` (\circledvee) 168
`\circledwedge` (\circledwedge) 168
`\circleft` (\circleft) 170
`\circright` (\circright) 170
`\cite` 88, 220, 224
`\citeform` 235, 236
`\citeleft` 235
`\citement` 235
`\citepunct` 235
`\citeright` 235
`\cleaders` 278
`\cleardoublepage` 254
`\clearpage` 87, 253
`\cline` 128, 175
`\clubsuit` (\clubsuit) 136
`\cmdruler` 178, 179
`\cmr` 328
`\cms` 328
`\cmtt` 328
`\Colonapprox` (\Colonapprox) 170
`\colonapprox` (\colonapprox) 170
`\Coloneq` (\Coloneq) 170
`\coloneq` (\coloneq) 170

$\backslash Coloneqq$ ($::=$) 170
 $\backslash coloneqq$ ($::=$) 170
 $\backslash Colonsim$ ($::\sim$) 170
 $\backslash colonsim$ ($::\sim$) 170
 $\backslash color$ 283
 $\backslash colorbox$ 283
 $\backslash columnsep$... 239, 245, 260, 263
 $\backslash columnseprule$... 239, 260, 263
 $\backslash columnwidth$ 260
 comment 環境 61
 $\backslash complement$ (\mathbb{C}) 168
 $\backslash cong$ (\cong) 134, 170
 $\backslash contentsname$ 259
 $\backslash coprod$ (\coprod) 134
 $\backslash cos$ (\cos) 122
 $\backslash cosh$ (\cosh) 122
 $\backslash cot$ (\cot) 122
 $\backslash coth$ (\coth) 122
 $\backslash cr$ 129
 $\backslash crcr$ 141
 $\backslash csc$ (\csc) 122
 $\backslash Cup$ (\cup) 165
 $\backslash cup$ (\cup) 134
 $\backslash curlyeqprec$ (\lesssim) 166
 $\backslash curlyeqsucc$ (\gtrsim) 166
 $\backslash curlyvee$ (\vee) 165
 $\backslash curlywedge$ (\wedge) 165
 $\backslash curvearrowleft$ (\curvearrowleft) 167
 $\backslash curvearrowright$ (\curvearrowright) 167
 $\backslash Cvs$ 324
 $\backslash cvs$ 323

D

$\backslash d$ (\AA) 61
 $\backslash dag$ (\dagger) 61
 $\backslash dagger$ (\dagger) 134
 $\backslash daleth$ (\daleth) 168
 dashjoin 環境 208
 $\backslash dashleftarrow$ (\dashleftarrow) 167, 170
 $\backslash dashrightarrow$ (\dashrightarrow) 170
 $\backslash dashline$ 208, 209
 $\backslash dashrightarrow$ (\dashrightarrow) 167, 170
 $\backslash dashv$ (\dashv) 134
 $\backslash date$ 45, 46
 $\backslash day$ 254
 $\backslash dbinom$ 164
 $\backslash ddag$ (\ddagger) 61
 $\backslash ddagger$ (\ddagger) 134
 $\backslash dddot$ (\ddot{a}) 161

$\backslash ddot$ (\ddot{a}) 161
 $\backslash \dot{a}$ (\dot{a}) 135
 $\backslash \dot{a}$ 161
 $\backslash \dot{a}$ 136
 $\backslash DeclareMathOperator$.. 150, 160, 161
 $\backslash DeclareRobustCommand$.. 93, 94
 $\backslash definecolor$ 283
 $\backslash deg$ (\deg) 122
 $\backslash Delta$ (Δ) 133
 $\backslash delta$ (δ) 132
 $\backslash deltaup$ (δ) 169
 $\backslash depthof$ 308
 description 環境 73
 $\backslash det$ (\det) 122
 $\backslash dfrac$ 164
 $\backslash DH$ (\mathbb{D}) 61
 $\backslash dh$ (δ) 61
 $\backslash diagdown$ (\searrow) 168
 $\backslash diagup$ (\nearrow) 168
 $\backslash Diamond$ (\diamond) 137, 169
 $\backslash diamond$ (\diamond) 134
 $\backslash Diamondblack$ (\blacklozenge) 169
 $\backslash Diamonddot$ (\diamond) 169
 $\backslash DiamonddotLeft$ ($\left\leftarrow\rightleftarrows$) 170
 $\backslash Diamonddotleft$ ($\left\leftarrow\rightleftarrows$) 170
 $\backslash DiamonddotRight$ ($\rightarrow\rightleftarrows$) 170
 $\backslash Diamonddotright$ ($\rightarrow\rightleftarrows$) 170
 $\backslash DiamondLeft$ ($\left\leftarrow\rightleftarrows$) 170
 $\backslash Diamondleft$ ($\left\leftarrow\rightleftarrows$) 170
 $\backslash DiamondRight$ ($\rightarrow\rightleftarrows$) 170
 $\backslash Diamondright$ ($\rightarrow\rightleftarrows$) 170
 $\backslash diamondsuit$ (\blacklozenge) 136
 $\backslash digamma$ (Γ) 168
 $\backslash dim$ (\dim) 122
 $\backslash displaybreak$ 163
 displaymath 環境 115
 $\backslash displaystyle$ 130, 164
 $\backslash div$ (\div) 134
 $\backslash divide$ 307
 $\backslash divideontimes$ (\oslash) 165
 $\backslash DJ$ (\mathbb{D}) 61
 $\backslash dj$ (\AA) 61
 $\backslash documentclass$.. 19, 24, 29
 $\backslash documentstyle$ 29
 document 環境 19
 $\backslash dot$ (\dot{a}) 135
 $\backslash \dot{a}$ 161
 $\backslash doteq$ (\doteq) 134, 170
 $\backslash doteqdot$ (\doteqdot) 166
 $\backslash dotfill$ 147, 278

$\backslash dotplus$ ($\dot{+}$) 165
 $\backslash dotsb$ 154
 $\backslash dotsc$ 154
 $\backslash dotsi$ 154
 $\backslash dotsm$ 154
 $\backslash dotso$ 154
 dottedjoin 環境 208
 $\backslash dottedline$ 208, 209
 $\backslash doublebarwedge$ ($\overline{\wedge}$) .. 165
 $\backslash doublebox$ 270
 $\backslash doublespacing$ 55
 $\backslash Downarrow$ (\Downarrow) ... 125, 135
 $\backslash downarrow$ (\downarrow) ... 125, 135
 $\backslash downbracefill$ 278
 $\backslash downdownarrows$ (\Downarrow) .. 167
 $\backslash downharpoonleft$ (\leftleftarrows) .. 167
 $\backslash downharpoonright$ ($\rightarrow\rightleftarrows$) .. 167
 drawjoin 環境 208
 $\backslash drawline$ 208, 209

E

$\backslash E$ 202
 $\backslash ell$ (ℓ) 136
 $\backslash ellipse*$ 210
 $\backslash em$ 59, 60
 $\backslash emph$ 59, 60, 65
 $\backslash emptyset$ (\emptyset) 136
 $\backslash end$ 20
 $\backslash endfirsthead$ 181
 $\backslash endfoot$ 181
 $\backslash endhead$ 181
 $\backslash endlastfoot$ 181
 $\backslash EndPicture$ 43
 $\backslash endSetLabels$ 201
 $\backslash enskip$ 274
 $\backslash enspace$ 274
 $\backslash ensuremath$ 256
 enumerate 環境 .. 73, 74, 87
 $\backslash epsilon$ (ϵ) 132
 $\backslash epsilonup$ (ϵ) 169
 $\backslash eqcirc$ (\equiv) 166
 $\backslash Eqcolon$ (\equiv) 170
 $\backslash eqcolon$ (\equiv) 170
 $\backslash eqnarray*$ 環境 116
 $\backslash eqnarray$ 環境 109, 117
 $\backslash Eqqcolon$ (\equiv) 170
 $\backslash eqqqcolon$ (\equiv) 170
 $\backslash eqslantgtr$ (\gtrsim) 166
 $\backslash eqslantless$ (\lesssim) 166
 $\backslash equal$ 309, 310
 $\backslash equation$ 109
 equation 環境 115
 $\backslash equiv$ (\equiv) 134
 $\backslash eta$ (η) 132

\etaup (η) 169
 \eth (\eth) 168
 \evensidemargin .. 239, 321
 \exists (\exists) 136
 \exp (exp) 122

F

\fallingdotseq (\fallingdotseq) .. 166
 \fancyfoot 252
 \fancyhead 252
 \fancyput 273
 \fancyput* 273
 \fbox 266, 270, 273
 \fboxrule 266, 284
 \fboxsep 266, 284
 \fcolorbox 284
 figure*環境 .. 260, 261, 263
 \figurename 259
 figure 環境 .. 172, 173, 261, 263
 filecontents 環境 23
 \fill 277
 \filltype 210
 \fint (\int) 169
 \Finv (\Finv) 168
 flalign 環境 153, 155
 \flat (\flat) 136
 flushleft 環境 71, 72
 flushright 環境 71, 72
 \fnsymbol 108
 \footnote 59
 \footnotemark 177
 \footnotesize 78, 79
 \footnotetext 177
 \footrulewidth 251
 \footskip 239, 242, 244, 321
 \forall (\forall) 136
 \frac 123, 140, 144, 164
 \framebox 266, 270
 framebox 環境 273
 \frontmatter 52
 \frown (\frown) 134
 \fullwidth 241
 \verbatim 環境 272

G

\Game (\Game) 168
 \Gamma (Γ) 133
 \gamma (γ) 132
 \gammaup (γ) 169
 gathered 環境 162
 gather 環境 .. 152, 154, 162
 \gcd (gcd) 122
 \ge (\ge) 134

\genfrac 164
 \geometry 241
 \geqq (\geqq) 166
 \geqslant (\geqslant) 166
 \gg (\gg) 134
 \ggg (\ggg) 166
 \gimel (\gimel) 168
 \gnapprox (\gtrsim) 167
 \gneq (\gneq) 167
 \gneqq (\gneqq) 167
 \gnsim (\gnsim) 167
 \graphicspath 201
 \grave (\grave) 135
 \Grave 161
 \grid 208
 \GridLineWidth 202
 \gtfamily 81
 \gtapprox (\gtrapprox) 166
 \gtrdot (\gtrdot) 166
 \gtreqless (\gtrless) 166
 \gtreqqless (\gtrless) 166
 \gtrless (\gtrless) 166
 \gtrsim (\gtrsim) 166
 \gtfamily 81
 \guillemotleft (\ll) 61
 \guillemotright (\gg) 61
 \guilsinglleft (\langle) 61
 \guilsinglright (\rangle) 61
 \gvertneqq (\gtrless) 167

H

\H (\H) 61
 \hat (\hat) 135
 \Hat 161
 \hbar (\hbar) 136, 168
 \HCode 43
 \hdotsfor 153, 157
 \headfont 323
 \headheight .. 239, 242, 244, 321
 \headrulewidth 251
 \headsep . 239, 242, 244, 321
 \heartsuit (\heartsuit) 136
 \heightof 308
 \hfil 253, 277
 \hfill 200, 277
 \hline 128, 174, 175
 \hoffset 239, 245, 321
 \hom (hom) 122
 \hookleftarrow (\leftarrow) .. 135
 \hookrightarrow (\rightarrow) 135
 \hour 254
 \hphantom 143
 \href 293
 \hrulefill 278

\hslash (\hslash) 168
 \hspace 254
 \hspace* 254, 275
 \hss 141
 \Huge 78, 79
 \huge 78, 79
 \hypertarget 293
 \hyphenation 67

I

\i (i) 61
 \idotsint ($\int \cdots \int$) . 161, 169
 \iffloatpage 251
 \ifthenelse 309
 \iiint (\iiint) 161, 169
 \iint (\iint) 161, 169
 \Im (\Im) 136
 \imath (\imath) 136
 \in (\in) 134
 \include 279
 \includegraphics . 188, 190, 193, 196, 201
 \includeonly 279
 \indent 54
 indentation 環境 54
 \inf (inf) 122
 \infty (∞) 136
 \injlim (injlim) 160
 \input 198, 279
 \int (\int) 134
 \intercal (\intercal) 165
 \intertext 162
 \invamp (\wp) 168
 \iota (ι) 132
 \iotaup (ι) 169
 \isodd 309, 310
 \item 73
 itemize 環境 73
 \itshape 80

J

\j (j) 61
 \jmath (j) 136
 \Join (\Join) 137, 170
 \jput 208

K

\k (\k) 61
 \kappa (κ) 132
 \kappaup (κ) 169
 \ker (ker) 122

L

$\backslash L$ (\mathbb{L}) 61
 $\backslash 1$ ($\mathbb{1}$) 61
 $\backslash L$ 202
 $\backslash @chapter$ 325
 $\backslash label$ 88, 104, 112, 115, 173
 $\backslash labelenumi$ 75
 $\backslash Lambda$ (Λ) 133
 $\backslash lambda$ (λ) 132
 $\backslash lambdaubar$ ($\bar{\lambda}$) 169
 $\backslash lambda slash$ (λ) 169
 $\backslash lambda daup$ (λ) 169
 $\backslash langle$ (\langle) 125
 $\backslash LARGE$ 78, 79
 $\backslash Large$ 78, 79
 $\backslash large$ 78, 79, 325
 $\backslash LaTeX$ 71, 255
 $\backslash LaTeXE$ 255
 $\backslash layout$ 237
 $\backslash lbag$ (ℓ) 169
 $\backslash lbrace$ ($\{$) 125
 $\backslash lceil$ (\lceil) 125
 $\backslash ldots$ (\dots) 136
 $\backslash ldots$ 136
 $\backslash le$ (\leq) 134
 $\backslash leaders$ 278
 $\backslash leadsto$ (\rightsquigarrow) 137, 170
 $\backslash leadstoext$ (\rightsquigarrow) 170
 $\backslash left$ (\leq) 124
 $\backslash left($ 129
 $\backslash Leftarrow$ (\Leftarrow) 135
 $\backslash leftarrow$ (\leftarrow) 135
 $\backslash leftarrowfill$ 278
 $\backslash leftarrowtail$ (\leftarrow) 167
 $\backslash leftharpoonright$ (\leftarrow) 135
 $\backslash leftharpoonup$ (\leftarrow) 135
 $\backslash leftidx$ 122, 163
 $\backslash leftleftarrows$ (\Lleftarrow) 167
 $\backslash Leftrightarrow$ (\Leftrightarrow) 135
 $\backslash leftrightarrow$ (\leftrightarrow) 135
 $\backslash leftrightarrows$ (\Leftrightarrow) 167
 $\backslash leftrightharpoons$ (\Leftrightarrow) 167
 $\backslash leftright squigarrow$ (\Leftrightarrow) 167
 $\backslash leftsquigarrow$ (\rightsquigarrow) 170
 $\backslash leftthreetimes$ (λ) 165
 $\backslash lengthtest$ 309, 310
 $\backslash leqq$ (\leq) 166
 $\backslash leqslant$ (\leq) 166
 $\backslash lessapprox$ (\lesssim) 166
 $\backslash lessdot$ (\leq) 166
 $\backslash lesseqgtr$ (\lesseqgtr) 166
 $\backslash lesseqqtr$ (\lesseqqtr) 166

$\backslash lessgtr$ (\lesseqgtr) 166
 $\backslash lesssim$ (\lesssim) 166
 $\backslash lfloor$ (\lfloor) 125
 $\backslash lfoot$ 251
 $\backslash lgroup$ (\lgroup) 125
 $\backslash lhd$ (\langle) 137, 168
 $\backslash lhead$ 251
 $\backslash lim$ (\lim) 122
 $\backslash liminf$ (\liminf) 122
 $\backslash limits$ 123, 160
 $\backslash limsup$ (\limsup) 122
 $\backslash line$ 206
 $\backslash linewidth$ 200
 $\backslash Link$ 44
 $\backslash listfigurename$ 259
 $\backslash listfiles$ 26
 $\backslash listoffigures$ 50
 $\backslash listoftables$ 50
 $\backslash listtablename$ 259
 $\backslash lJoin$ (\Join) 170
 $\backslash ll$ (\ll) 134
 $\backslash llbracket$ (\llbracket) 169
 $\backslash llcorner$ (\llcorner) 168
 $\backslash Lleftarrow$ (\Lleftarrow) 167
 $\backslash lll$ (\lll) 166
 $\backslash lmoustache$ (\smile) 125
 $\backslash napprox$ ($\not\approx$) 167
 $\backslash neq$ (\neq) 167
 $\backslash neqq$ (\neqq) 167
 $\backslash lnsim$ (\sim) 167
 $\backslash log$ (\log) 122
 $\backslash Longleftarrow$ (\Longleftarrow) 135
 $\backslash longleftarrow$ (\longleftarrow) 135
 $\backslash Longleftrightarrow$ (\Longleftrightarrow) 135
 $\backslash longmappedfrom$ (\Longleftarrow) 170
 $\backslash longmappedfrom$ (\leftarrow) 170
 $\backslash Longmapsto$ (\Longrightarrow) 170
 $\backslash longmapsto$ (\longmapsto) 135
 $\backslash Longmmapsto$ (\Longmmapsto) 170
 $\backslash longmmapsto$ (\leftarrow) 170
 $\backslash Longmmapsto$ (\Longrightarrow) 170
 $\backslash longmmapsto$ (\longmapsto) 170
 $\backslash Longrightarrow$ (\Longrightarrow) 135
 $\backslash longrightarrow$ (\longrightarrow) 135
 $\backslash longtable$ 環境 181
 $\backslash looparrowleft$ (\looparrowleft) 167
 $\backslash looparrowright$ (\looparrowright) 167

$\backslash lozenge$ (\diamond) 168
 $\backslash rbox$ 環境 268
 $\backslash lrcorner$ (\lrcorner) 168
 $\backslash lrtimes$ (\rtimes) 170
 $\backslash Lsh$ (\Lsh) 167
 $\backslash lstdefineline$ 287
 $\backslash lstdefinestyle$.. 285, 286
 $\backslash lstinputlisting$ 285
 $\backslash lstlistingname$ 289
 $\backslash lstlisting$ 環境 285
 $\backslash lstlistoflistingname$.. 289
 $\backslash lstlistoflistings$ 289
 $\backslash lstnewenvironment$ 286
 $\backslash lstset$ 285
 $\backslash ltimes$ (\ltimes) 165
 $\backslash ltrans$ 122
 $\backslash lvertneqq$ (\leq) 167

M

$\backslash makeatletter$ 98, 100
 $\backslash makeatother$ 98, 100
 $\backslash makebox$ 207, 266
 $\backslash maketitle$.. 46, 48, 52, 176, 323
 $\backslash maltese$ (\Maltese) 168
 $\backslash Mappedfrom$ (\Leftrightarrow) 170
 $\backslash mappedfrom$ (\leftrightarrow) 170
 $\backslash Mapsto$ (\mapsto) 170
 $\backslash mapsto$ (\mapsto) 135
 $\backslash marginpar$ 59
 $\backslash marginparpush$ 239, 321
 $\backslash marginparsep$ 239, 242, 245, 321
 $\backslash marginparwidth$.. 239, 242, 245, 321
 $\backslash markboth$ 250
 $\backslash markright$ 250
 $\backslash MARU$ 75
 $\backslash mathbb$ 118
 $\backslash mathbf$ 118, 142
 $\backslash mathcal$ 118
 $\backslash mathfrak$ 118
 $\backslash mathit$ 2, 118
 $\backslash mathnormal$ 118
 $\backslash mathrel$ 166
 $\backslash mathrm$ 118, 133
 $\backslash mathsf$ 118
 $\backslash mathstrut$ 143
 $\backslash mathhtt$ 118
 $\backslash math$ 環境 114
 $\backslash matrix$ 128
 $\backslash matrixput$ 208
 $\backslash matrix$ 環境 .. 128, 153, 156, 157

`\ntwoheadleftarrow` (\Lleftarrow) 170
`\ntwoheadrightarrow` (\Rrightarrow) 170
`\nu` (ν) 132
`\null` 323
`\number` 255
`\numberwithin` 159
`\nuup` (ν) 169
`\nvarparallel` ($\#$) 170
`\nvarparallelinv` ($\#$) 170
`\nVDash` ($\#$) 167
`\nvDash` ($\#$) 167
`\nvdash` ($\#$) 167
`\Nwarrow` ($\#$) 170
`\narrow` ($\#$) 135

O

`\O` (\O) 61
`\o` (\o) 61
`\oddsidemargin` 239, 321
`\odot` (\odot) 134
`\OE` (\O) 61
`\oe` (\o) 61
`\oiint` (\oiint) 169
`\oiintclockwise` (\oiint) 169
`\oiintctrlockwise` (\oiint) 169
`\ooint` (\oint) 169
`\oointclockwise` (\oint) 169
`\oointctrlockwise` (\oint) 169
`\oint` (\oint) 134, 161
`\ointclockwise` (\oint) 169
`\ointctrlockwise` (\oint) 169
`\Omega` (Ω) 133
`\omega` (ω) 132
`\omegaup` (ω) 169
`\ominus` (\ominus) 134
`\onecolabstract` 313
`\onecolumn` 260, 261
`\onehalfspacing` 55
`\oalign` 141
`\openJoin` (\times) 170
`\opentimes` (\times) 170
`\operatorname` 161
`\oplus` (\oplus) 134
`\or` 311
`\oslash` (\oslash) 134
`\otimes` (\otimes) 134
`\oval` 206

`\Ovalbox` 270
`\ovalbox` 270
`\overbrace` 135
`\overleftarrow` 135, 163
`\overleftrightharrow` 163
`\overline` 135
`\overrightarrow` 135, 163
`\overset` 163

P

`\P` (\P) 61
`\p@` 321
`\pagecolor` 283
`\pagenumbering` 250
`\pageref` 105, 111, 112
`\pagestyle` 249
`\paperheight` 242
`\paperwidth` 242
`\par` 53, 68, 70
`\paragraph` 48, 49
`\parallel` (\parallel) 134
`\parbox` 267
`\parindent` 53, 68
`\parskip` 54
`\part` 48, 49
`\partial` (∂) 136
`\path` 210, 292
`\Perp` (\perp) 170
`\perp` (\perp) 134
`\phantom` 143
`\Phi` (Φ) 133
`\phi` (ϕ) 132
`\phiiup` (ϕ) 169
`\Pi` (Π) 133
`\pi` (π) 132
`\picsquare` 208
`\Picture` 43
`\Picture+` 43
`picture` 環境 172, 186, 205, 211
`\pitchfork` (\pitchfork) 166
`\piup` (π) 169
`\pLaTeX` 255
`\pLaTeXe` 255
`\pm` (\pm) 134
`\pmatrix` 128
`pmatrix` 環境 128, 156
`\pmod` 122
`\postchaptername` 259
`\postpartname` 259
`\postsectionname` 257
`\pounds` (\pounds) 61
`\Pr` (\Pr) 122
`\prec` (\prec) 134

`\precapprox` (\preccurlyeq) 166
`\preccurlyeq` (\preccurlyeq) 166
`\preceq` (\preceq) 134
`\preceqq` (\preceqq) 170
`\prechaptername` 259
`\precnapprox` (\preccurlyeq) 167
`\precnsim` (\preccurlyeq) 167
`\precsim` (\preccurlyeq) 166
`\prepartname` 259
`\presectionname` 257
`\prime` (\prime) 136
`\prod` (\prod) 134
`\projlim` ($\text{\text{projlim}}$) 160
`\propto` (\propto) 134
`\protect` 95
`\providecommand` 93
`\ps@total` 253
`\Psi` (Ψ) 133
`\psi` (ψ) 132
`\psiiup` (ψ) 169
`\pTeX` 255
`\put` 206
`\putfile` 208

Q

`\qbezier` 206, 210, 211
`\quad` 120, 274
`\quadr` 120, 274
`quotation` 環境 64
`\quotedblbase` ($\text{\text{quotedblbase}}$) 61
`\quotesinglbase` ($\text{\text{quotesinglbase}}$) 61
`quote` 環境 54, 64

R

`\r` (\r) 61
`\R` 202
`\raggedleft` 72
`\raggedright` 72, 324
`\raisebox` 269
`\rangle` (\rangle) 125
`\rbag` ($\text{\text{rbag}}$) 169
`\rbrace` ($\}$) 125
`\rceil` ($\text{\text{rceil}}$) 125
`\Re` (\Re) 136
`\real` 308
`\ref` 88, 105, 112, 173
`\reflectbox` 192
`\refname` 257
`\refstepcounter` 108
`\renewcommand` 92, 93
`\renewenvironment` 93
`\Requirepackage` 321
`\resizebox` 193
`\rfloor` (\lfloor) 125

- `\rfoot` 251
`\rgroup` (\jmath) 125
`\rhd` (\triangleright) 137, 168
`\rhead` 251
`\rho` (ρ) 132
`\rhoup` (ρ) 169
`\right` 124
`\right)` 129
`\Rightarrow` (\Rightarrow) 135
`\rightarrow` (\rightarrow) 135
`\rightarrowfill` 278
`\rightarrowtail` (\rightarrowtail) 167
`\rightharpoondown` (\curvearrowright) ..
135
`\rightharpoonup` (\curvearrowright) .. 135
`\rightleftarrows` (\Leftrightarrow) 167
`\rightleftharpoons` (\rightleftharpoons) ..
135, 167
`\rightarrowtail` (\rightarrowtail) ...
167
`\rightsquigarrow` (\rightsquigarrow) 167
`\righthreetimes` (\times) 165
`\risingdotseq` ($\dot{=}$) 166
`\rJoin` (\Join) 170
`\rmfamily` 80
`\rmoustache` (\hookrightarrow) 125
`\Roman` 74, 108
`\roman` 74, 108
`\rotatebox` 192
`\rrbracket` (\rrbracket) 169
`\Rsh` (\Rsh) 167
`\rtimes` (\rtimes) 165
`\rule` 269
- S**
- `\S` (\S) 61
`\samepage` 254
`\savebox` 268
`\sb` 121
`\Sbox` 271
`\sbox` 268
`Sbox` 環境 270
`\scalebox` 192
`\scriptscriptstyle` ... 130, 164
`\scriptsize` 78, 79
`\scriptstyle` 130, 164
`\scshape` 80
`\searrow` (\searrow) 170
`\searrow` (\searrow) 135
`\sec` (sec) 122
`\section` ... 48, 49, 109, 318, 324
`\sectionmark` 252
- `\setboolean` 309
`\setbox` 268
`\setcounter` .. 107, 108, 307
`\SetLabels` 201
`\setlength` 263, 307
`\setminus` 134
`\settoheight` 265, 307
`\settoheight` 307
`\settoheight` 265
`\settowidth` 265, 307
`\sffamily` 80
`\sfrac` 131
`\shadowbox` 270
`\sharp` (\sharp) 136
`\shortmid` (\shortmid) 166
`\shortparallel` (\shortparallel) ... 166
`\shoveleft` 153, 155
`\shoveright` 153, 155
`\ShowGrid` 201, 202
`\sideset` 163
`\Sigma` (Σ) 133
`\sigma` (σ) 132
`\sigmaup` (σ) 169
`\sim` (\sim) 134
`\simeq` (\simeq) 134
`\sin` (sin) 122
`\singlespacing` 55
`\sinh` (sinh) 122
`\slshape` 80
`\small` 78, 79
`\smallfrown` (\smallfrown) 166
`\smallmatrix` 157
`\smallsetminus` (\setminus) .. 165
`\smallskip` 276
`\smallsmile` (\smallsmile) 166
`\smash` 143
`\smile` (\smile) 134
`\sp` 121
`spacing` 環境 55
`\spadesuit` (\spadesuit) 136
`\spbreve` 161
`\spcheck` 161
`\spddot` 161
`\spddot` 161
`\spdot` 161
`\special` 213
`\sphat` 161
`\sphericalangle` (\sphericalangle) . 168
`\spline` 210
`split` 環境 153, 154
`\sptilde` 161
`\sqcap` (\sqcap) 134
`\sqcapplus` (\sqcapplus) 168
`\sqcup` (\sqcup) 134
- `\sqcupplus` (\sqcupplus) 168
`\sqqintop` ($\int\!\!\int$) 169
`\sqqintop` ($\int\!\!\int$) 169
`\sqtint` (\int) 169
`\sqrt` 123
`\sqsubset` (\sqsubset) 137, 166
`\sqsubseteq` (\sqsubseteq) 134
`\sqsupset` (\sqsupset) 137, 166
`\sqsupseteq` (\sqsupseteq) 134
`\square` (\square) 168
`\SS` (SS) 61
`\ss` (\ss) 61
`\stackrel` 141
`\star` (\star) 134
`\stepcounter` 108
`\stretch` 277
`\strictfi` (ϵ) 170
`\strictif` (ϵ) 170
`\strictiff` (ϵ) 170
`\strut` 201
`subarray` 環境 157, 158
`subequations` 環境 158
`\subparagraph` 48, 49
`\subsection` 48, 49, 318, 322
`\Subset` (\Subset) 166
`\subset` (\subset) 134
`\subseteq` (\subseteq) 134
`\subseteqeq` (\subseteqeq) 166
`\subsetneq` (\subsetneq) 167
`\subsetneqq` (\subsetneqq) 167
`\substack` 141, 157, 158
`\subsubsection` . 48, 49, 318
`\succ` (\succ) 134
`\succapprox` (\succapprox) 166
`\succcurlyeq` (\succcurlyeq) 166
`\succeq` (\succeq) 134
`\succeqq` (\succeqq) 170
`\succnapprox` (\succnapprox) 167
`\succnsim` (\succnsim) 167
`\succsim` (\succsim) 166
`\sum` (Σ) 134
`\sup` (sup) 122
`\Supset` (\Supset) 166
`\supset` (\supset) 134
`\supseteq` (\supseteq) 134
`\supseteqeq` (\supseteqeq) 166
`\supsetneq` (\supsetneq) 167
`\supsetneqq` (\supsetneqq) 167
`\surd` (\surd) 136
`\Swarrow` (\swarrow) 170
`\swarrow` (\swarrow) 135
`\symbol` 165
`\syntaxonly` 82

T

- \t (oo) 61
- \T 202
- tabbing 環境 173
- \table 109
- table*環境 ... 260, 261, 263
- tablename 259
- \tableofcontents .. 50, 257
- table 環境 ... 172, 173, 181, 261, 263
- tabular*環境 87
- tabularx 環境 184, 185
- tabular 環境 .. 87, 172, 173, 176, 179, 180, 184
- \tag 153, 158
- \tag* 153
- \tan (tan) 122
- \tanh (tanh) 122
- \tau (τ) 132
- \tauup (τ) 169
- \tbinom 164
- \TeX 71, 255
- \textorpdfstring .. 294, 295
- \text 118, 119, 150
- \textacutedbl (") 258
- \textasciiacute (') .. 258
- \textasciibreve (˘) .. 258
- \textasciicaron (ˇ) .. 258
- \textasciicircum 12
- \textasciidieresis (¨) ... 258
- \textasciigrave (˘) .. 258
- \textasciimacron (˘) 258
- \textasciitilde 12
- \textasteriskcentered (*) 258
- \textbackslash 12
- \textbaht (฿) 258
- \textbar 12
- \textbardbl (||) 258
- \textbf 80
- \textbigcircle (○) .. 258
- \textblank (b) 258
- \textborn (★) 258
- \textbrokenbar (|) 258
- \textbullet (●) 258
- \textcelsius (°C) 258
- \textcent (¢) 258
- \textcentoldstyle (c) 258
- \textcircledP (Ⓟ) .. 258
- \textcolonmonetary (C) .. 258
- \textcolor 283
- \textcopyright (©) ... 258
- \textcopyright (©) .. 258
- \textcurrency (¤) ... 258
- \textdagger (†) 258
- \textdaggerdbl (‡) ... 258
- \textdblhyphen (=) ... 258
- \textdblhyphenchar (=) ... 258
- \textdegree (°) 258
- \textdied (+) 258
- \textdiscount (%) 258
- \textdiv (÷) 258
- \textdivorced (o|o) 258
- \textdollar (\$) 258
- \textdollaroldstyle (\$) . 258
- \textdong (₫) 258
- \textdownarrow (↓) ... 258
- \texteightoldstyle (8) ... 258
- \textestimated (€) ... 258
- \texteuro (€) 258
- \textfiveoldstyle (5) 258
- \textflorin (f) 258
- \textfouroldstyle (4) 258
- \textfractionsolidus (/) 258
- \textgravedbl (¨) 258
- \textgreater 12
- \textgt 81
- \textguarani (₲) 258
- \textheight .. 239, 242, 243, 245, 321
- \textinterrobang (‡) . 258
- \textinterrobangdown (‡) 258
- \textit 60, 80
- \textlangle (⟨) 258
- \textlbracketdbl (||) ... 258
- \textleaf (♻) 258
- \textleftarrow (←) .. 258
- \textless 12
- \textlira (₺) 258
- \textlnot (¬) 258
- \textliquill (℥) 258
- \textmarried (∞) 258
- \textmc 81
- \textmd 80
- \textmho (℧) 258
- \textminus (−) 258
- \textmtt 63
- \textmu (μ) 258
- \textmusicalnote (♯) . 258
- \textnaira (₦) 258
- \textnineoldstyle (9) 258
- \textnumero (№) 258
- \textohm (Ω) 258
- \textonehalf (½) 258
- \textoneoldstyle (1) . 258
- \textonequarter (¼) .. 258
- \textonesuperior (¹) . 258
- \textopenbullet (◦) .. 258
- \textordfeminine (ª) . 258
- \textordmasculine (º) 258
- \textparagraph (¶) ... 258
- \textperiodcentered (·) .. 258
- \textpertousand (‰) ... 258
- \textperthousand (‰) ... 258
- \textpeso (₱) 258
- \textpilcrow (¶) 258
- \textpm (±) 258
- \textquotedbl (") 61
- \textquotesingle (') . 258
- \textquoteststraightbase (.) 258
- \textquoteststraightdblbase (¨) 258
- \texttriangle (∇) 258
- \texttrbracketdbl (||) ... 258
- \textrecipe (℞) 258
- \textreferencemark (*) .. 258
- \textregistered (®) . 258
- \textrightarrow (→) . 258
- \textrm 80
- \texttrquill (}) 258
- \textsc 80
- \textsection (§) 258
- \textservicemark (SM) 258
- \textsevenoldstyle (7) ... 258
- \textsf 80
- \textsixoldstyle (6) . 258
- \textsl 80
- \textsterling (£) 258
- \textstyle 130, 164
- \textsurd (√) 258
- \textthreeoldstyle (3) ... 258
- \textthreequarters (¾) ... 258
- \textthreequartersemdash (—) 258
- \textthreesuperior (³) ... 258
- \texttildelow (˘) 258

`\texttimes` (\times) 258
`\texttrademark` ($^{\text{TM}}$) ... 258
`\texttt` 63, 80
`\texttwelveudash` (--) 258
`\texttwooldstyle` (2) . 258
`\texttwo superior` (2^{sup}) . 258
`\textuparrow` (\uparrow) 258
`\textwidth` ... 239, 241, 242,
245, 260, 321
`\textwon` (W) 258
`\textyen` (Y) 258
`\textzerooldstyle` (0) 258
`\tfrac` 164
`\TH` (P) 61
`\th` (b) 61
`\thanks` 47
`\the` 101
`thebibliography` 環境 .. 22,
220, 257
`theequation` 159
`theindex` 環境 22
`theorem` 137, 138
`theorembodyfont` 138
`theoremheaderfont` 138
`theoremstyle` 138
`\thepage` 253
`\therefore` (\therefore) 166
`\TheSbox` 270
`\thesection` 109
`\Theta` (Θ) 133
`\theta` (θ) 132
`\thetaau` (θ) 169
`\thickapprox` (\approx) 166
`\Thicklines` 209, 210
`\thicklines` 207, 210
`\thickmuskip` 119
`\thicksim` (\sim) 166
`\thickspace` 162
`\thinlines` 207
`\thinmuskip` 119
`\thinspace` 162, 274
`\thisfancyput` 273
`\thisfancyput*` 273
`\thispagestyle` 250
`\tilde` (\tilde{a}) 135
`\Tilde` 161
`\times` (\times) 134
`\tiny` 78, 79
`\title` 45, 46
`\titlepage` 323
`\today` 254
`\Top` (\T) 169
`\top` (\T) 136
`\topmargin` 239, 321

`\toprule` 178
`\topskip` 243
`\triangle` (Δ) 136
`\triangledown` (∇) 168
`\triangleleft` (\triangleleft) 134
`\trianglelefteq` (\trianglelefteq) . 166
`\triangleq` (\triangleq) 166
`\triangleright` (\triangleright) ... 134
`\trianglerighteq` (\trianglerighteq) 166
`\ttfamily` 63, 80
`\two@digits` 255
`\twocolumn` 260, 261, 313
`\twoheadleftarrow` (\twoheadleftarrow) ...
167
`\twoheadrightarrow` (\twoheadrightarrow) .
167

U

`\u` (i) 61
`\ulcorner` (\ulcorner) 168
`\unboldmath` 142
`\underbrace` 135
`\underleftarrow` 163
`\underlefrightharpoonup` . 163
`\underline` 135, 269
`\underrightarrow` 163
`\underset` 163
`\unlhd` (\unlhd) 137, 168
`\unrhd` (\unrhd) 137, 168
`\Uparrow` (\Uparrow) 125, 135
`\uparrow` (\uparrow) 125, 135
`\upbracefill` 278
`\Updownarrow` (\Updownarrow) 125, 135
`\updownarrow` (\updownarrow) . 125, 135
`\upharpoonleft` (\upharpoonleft) ... 167
`\upharpoonright` (\upharpoonright) .. 167
`\uplus` (\uplus) 134
`\Upsilon` (Υ) 133
`\upsilon` (υ) 132
`\upsilonup` (υ) 169
`\upuparrows` (\upuparrows) 167
`\urcorner` (\urcorner) 168
`\url` 233, 292, 293
`\urlstyle` 292
`\usebox` 268
`\usefont` 165
`\usepackage` 24, 86, 323

V

`\v` (a) 61
`\value` 108
`\varclubsuit` (\clubsuit) 169
`\varDelta` (Δ) 160
`\vardiamondsuit` (\blacklozenge) .. 169

`\varepsilon` (ε) 133
`\varepsilonup` (ε) 169
`\varg` (g) 169
`\varGamma` (Γ) 160
`\varheartsuit` (\heartsuit) 169
`\varinjlim` (\varinjlim) 160
`\varkappa` (\varkappa) 168
`\varLambda` (Λ) 160
`\varliminf` (\varliminf) 160
`\varlimsup` (\varlimsup) 160
`\varnothing` (\emptyset) 168
`\varoiintclockwise` (\varoiint)
169
`\varoiintctrlockwise`
(\varoiint) 169
`\varointclockwise` (\varoint) ..
169
`\varointctrlockwise`
(\varoint) 169
`\varointclockwise` (\varoint) 169
`\varointctrlockwise` (\varoint)
169
`\varOmega` (Ω) 160
`\varparallel` (\parallel) 170
`\varparallelinv` (\parallel) .. 170
`\varPhi` (Φ) 160
`\varphi` (φ) 133
`\varphiup` (φ) 169
`\varPi` (Π) 160
`\varpi` (ϖ) 133
`\varpiup` (ϖ) 169
`\varprod` (\prod) 169
`\varprojlim` (\varprojlim) 160
`\varpropto` (\propto) 166
`\varPsi` (Ψ) 160
`\varrho` (ϱ) 133
`\varrhoup` (ϱ) 169
`\varSigma` (Σ) 160
`\varsigma` (ς) 133
`\varsigmaup` (ς) 169
`\varspadesuit` (\spadesuit) 169
`\varsubsetneq` (\subsetneq) 167
`\varsubsetneqq` (\subsetneqq) .. 167
`\varsupsetneq` (\supsetneq) 167
`\varsupsetneqq` (\supsetneqq) .. 167
`\varTheta` (Θ) 160
`\vartheta` (ϑ) 133
`\varthetaup` (ϑ) 169
`\vartriangle` (Δ) 168
`\vartriangleleft` (\triangleleft) 166
`\vartriangleright` (\triangleright) ...
166

`\varUpsilon` (Υ) 160
`\varv` (v) 169
`\varw` (w) 169
`\varXi` (Ξ) 160
`\vary` (y) 169
`\VDash` (\Vdash) 170
`\vDash` (\Vdash) 166
`\vDash` (\Vdash) 166
`\vdash` (\vdash) 134

`\vdots` (\vdots) 136
`\vec` (\vec{a}) 135
`\Vec` 161
`\vector` 206
`\vee` (\vee) 134
`\veebar` (\veebar) 165
`\verb` 62, 63, 272
`\VerbatimEnvironment` . 273
`Verbatim` 環境 273
`verbatim` 環境 . 62, 88, 203,
272, 273, 333
`\VerbBox` 272
`\Vert` (\parallel) 125

`\vert` ($|$) 125
`\vline` 128, 174, 175
`Vmatrix` 環境 156
`vmatrix` 環境 156
`\voffset` 239, 245, 321
`\vphantom` 143
`\vskip` 323, 324
`\vspace` 254
`\vspace*` 254, 276
`\VvDash` (\Vdash) 170
`\Vvdash` (\Vdash) 166

W

`\wedge` (\wedge) 134
`\whiledo` 309
`\widehat` 135
`\widetilde` 135
`\widthof` 308
`\wp` (\wp) 136
`\Wr` (\wr) 168
`\wr` (\wr) 134

X

`\Xi` (Ξ) 133
`\xi` (ξ) 132
`\xiup` (ξ) 169
`\xleaders` 278
`\xleftarrow` 163
`\xrightarrow` 163
`\xspace` 281

Y

`\year` 254
`\yen` (\yen) 168

Z

`\zeta` (ζ) 132
`\zetaup` (ζ) 169

せ

`\西曆` 254

わ

`\和曆` 254

索引

数字/記号

□ 60, 99
 # 11, 99
 \$ 11, 15, 99, 114
 コンソールの—— iii
 \$\$ 114
 \$texmf 19
 % 11, 61, 99
 & 11, 99
 array 環境の—— 126
 eqnarray* 環境の——
 116
 tabular 環境の—— 174
 ’ 63
 (..... 125
) 125
 * 17
 , (読点) 57
 - 66
 ハイフンとしての—— .. 66
 マイナスとしての—— .. 66
 -- 66
 --- 66
 . (句点) 57
 ... (中点 3 点リーダー) .. 136
 ... (下付 3 点リーダー) .. 136
 / 125, 144
 区切り記号の—— 125
 分数の—— 130
 : 58
 ; 58
 < 11
 = 56
 eqnarray 環境中の—— .
 116
 代入としての—— 56
 等号としての—— 114
 > 11
 @ 98
 [..... 125
 \ 11, 91, 99
] 125

^ 11, 99, 121
 _ 11, 99, 121
 ‘ 63
 { 11, 99
 } 11, 99
 | 11, 125
 ~ 11, 70, 99, 275
 10pt 56, 85, 86
 10 進数 32
 11pt 85, 86
 12pt 85, 86
 12Q 86
 14pt 86
 14Q 86
 16 進数 32
 16 ビットエンコーディング . 32
 17pt 86
 1 段組 85
 20pt 86
 21pt 86
 25pt 86
 2 次ベジェ曲線 211
 2 段組 83, 85, 253
 ——の段間の罫線 .. 259
 ——のときの段間 .. 259
 30pt 86
 36pt 86
 3 次ベジェ曲線用 211
 3 点リーダー 136
 下付き—— 136
 中点—— 136
 43pt 86
 9pt 86

A

a0paper 241
 a1paper 241
 a2paper 241
 a3paper 241
 a4j 85, 86
 a4paper 85, 86, 241, 294
 a4var 86

a5j 85, 86
 a5paper 85, 86, 241, 294
 a6paper 86, 241
 abbrev 230
 Abrahams, Paul W. 348
 abstract 313
 abstract 48, 52
 Acrobat Reader 32
 address (BibTeX) 230
 Adobe 38
 Adobe Acrobat 197–199
 Adobe Reader .. 30, 32, 338
 Adobe-Japan1-6 332
 afterpage 87
 Aleph 332
 alltt 62
 alpha 230
 Alver, Morten 236
 American Mathematical
 Society 87, 150
 amount of substance 57
 ampere 57
 amsbsy 142, 150
 amscd 150, 159
 amsfontr 136, 150, 165
 amsgen 150
 \LaTeX 87, 150
 amsmath . 87, 118, 128, 142,
 150, 161, 162, 350
 amsopn 150
 amssymb 118, 136, 150, 165,
 198
 \LaTeX 87, 150
 amstext 150
 amsthm 138
 amxtra 150, 161
 annote (BibTeX) 230
 APT 336
 array 87
 Arseneau, Donald . 234, 292
 article 48, 84, 313
 article (文献の種類) 229

article.cls 84
 ASCII 333
 asmcld 150
 asymmetric 244
 ATSUI 327
 author (BIBTEX) . 223, 230,
 233
 author 46
 .aux (拡張子) 10, 22, 50, 225
 avant 328

B

b0paper 241
 b1paper 241
 b2paper 241
 b3paper 241
 b4j 85, 86
 b4paper 85, 86, 241
 b5j 85, 86
 b5paper 85, 86, 241, 294
 b5var 86
 b6paper 241
 babel 87, 348
 backgroundcolor 290
 balance 260
 bash 339
 basicstyle 289
 Batada, Nizar 236
 .bb (拡張子) 23
 .bb1 (拡張子) 22, 224
 Beebe, Nelson v, 338
 Berliner, Brian 307
 Berners-Lee, Tim 89
 Berry, Karl 38, 350
 .bib (拡張子) 22
 bib2xml 338
 BIBTEX 219

BIBTEX

address 230
 annote 230
 author ... 223, 230, 233
 booktitle 230
 chapter 230
 crossref 230
 edition 230, 232
 editor 229, 230
 howpublished 230, 233
 journal 230, 232
 key 230
 month 230, 233
 note 223, 230
 number 230, 232
 organization 230
 pages 230, 232

publisher 223, 230, 232
 school 230
 series 230, 232
 title 223, 230
 volume 230, 232
 year 223, 230
 yomi 223, 232
 BIBTEX2HTML 338
 bindingoffset 244
 bk10.clo 84
 bk11.clo 84
 bk12.clo 84
 .blg (拡張子) 22
 bm 87, 142
 bmargin 243
 BMP 189
 .bmp (拡張子) 23, 30
 body 242
 book 84
 book (文献の種類) 229, 232
 book.cls 19, 84
 booklet (文献の種類) 229
 bookman 328
 bookmarks 294
 bookmarksnumbered ... 294
 bookmarksopen 294
 bookmarkstype 294
 booktabs 178
 booktitle (BIBTEX) ... 230
 boookmarksopenlevel ... 294
 bottom 243
 Braams, Johannes v, 62, 347
 breaklines 288
 breaklinks 294
 .bst (拡張子) 22

C

Calc 185
 calc .. 87, 241, 271, 307, 308
 Calc2LATEX 185
 Calishain, Tara 341
 Cameron, Debra 340
 candela 57
 caption 289
 captionpos 289
 Carlisle, David .. v, 74, 112,
 129, 179, 181, 184,
 282, 309, 347
 cd 12, 13
 centering 244
 centertags 151
 ceqno 151
 Chaikin 曲線 210
 chapter (BIBTEX) 230

chapter (カウンタ) 106
 chapter 46, 49
 character 46, 76
 charset 41
 charter 328
 Chen, Pehong v
 Cheng, William Chia-Wei ..
 199, 338
 Chicago Style 345
 Chinese 32
 Christiansen, Tom 340
 CID Type0 36
 cid-x.map 33
 CID フォント 32
 cite 234, 235
 class 82
 classes 84
 classes.dtx 84
 classes.ins 84
 .clo (拡張子) 22
 .cls (拡張子) 21
 cmtt 63
 Cochran, Steven Douglas ...
 200
 Collins, John 306
 color 87, 215, 281, 282
 colorlinks 294
 columnsep 245
 comment 61
 commentstyle 289
 Computer Modern .. 81, 328
 conference 233
 config.ps 39
 Contents 50
 contents 45, 48
 ConTeXt 37
 copy 12
 Courier 330
 courier 328
 cp 13
 CreateBB 23
 crossref (BIBTEX) 230
 CSS i
 CTAN 346
 CVS 307

D

date 46
 dcolumn 87, 179
 del 12
 delarray 87, 129
 delimiter array 129
 DeTeX 338
 Dia 338

dir 12
divide 243
 DocStrip 349
 DocStrip 84
 document 45
 Doerr, Hans-Peter 186
 Doob, Michael 348
 Dornfest, Rael 341
 doublespace 55
 Dougherty, Dale 340
 Downes, Michael v
draft 85, 202
 Drakos, Nikos 338
.dtx (拡張子) 21
 DuBois, Paul 339
 Duchier, Denys 349
 Duggan, Angus 39
 DVI 7, 11, 29
.dvi (拡張子) 22, 28
 dviout 17
 Dvipdfm 30, 245
dvipdfm 187, 245, 294
Dvipdfmx 30, 216, 338
dvipdfmx 187, 188, 196
dvipdfmx.def 187
dvips 188, 196, 245, 294
dvips 38, 39, 215, 245
dvipsnames 282

E

Easy \TeX 27
 Eaton, John 217, 338
 ebb 23, 189
eclpsf.sty 337
 EC フォント 348
 edition (B \TeX) 230, 232
 editor (B \TeX) 229, 230
 eepic 186, 198, 209
 ehis 337
 Eijkhout, Victor 61
 electric current 57
 elisp 337, 351
 em-dash 66
 Emacs .. 8, 26, 296, 298, 337
 VC 307
 emath 352
 emphasis 59
 en-dash 66
english 86, 333
 enumerate 74, 75, 87
 enumi (カウンタ) 106
 enumii (カウンタ) 106
 enumiii (カウンタ) 106
 enumiv (カウンタ) 106

EPDF 186, 333
.epdf (拡張子) 30
 epic 186, 198, 208, 209
.eps (拡張子) 22, 30
epsbox.sty 337
 epstopdf 194, 216
 equation (カウンタ) .. 106,
 159
errortest.tex 14, 16
 Esser, Thomas 337
 ϵ - \TeX 327
 eucal 151
 Excel 185, 186
 Excel2tabular 185
 exceltex 186
executivepaper 241
expert 335

F

fancybox 270, 273
fancyhdr 250, 350
fancyheadings 250
.fd (拡張子) 22
 Fear, Simon 178
 Fedora Core 17
 —への導入 5
figure (カウンタ) 106
figure 172
fil 276
 file 189
fill 276
 Filiâtre, Jean-Christophe ...
 338
final 85, 202
firstline 288
firstnumber 289
fleqn 85, 115, 151
 float 172
 float 289
fn-in 41
 Fogel, Karl 340
fontdef.dtx 278
fontenc 60, 256, 330
fonts+ 41
foot 244
footnote (カウンタ) 106
footskip 244
 Foxit Reader 30
 Foxit Software Company 30
frame 290
framerule 290
framesep 290
 Friedl, Jeffrey E. F. 340
ftnright 87

G

gbm 34
gbmv 34
 geometry . 240, 241, 245, 247
 Ghostscript 33, 38
gif 41
 giga 57
 GIMP 338, 342
 GNU
 —Emacs 296
 —Ghostscript 33
 —FDL vi
 —GPL 198
gnu-head.pdf 191
 Gnuplot .. 26, 217, 338, 341,
 342, 351, 352
 Google 346
 Goossens, Michel iv, 342
 GothicBBB-Medium 34
 GPL 198
 Grätzer, George ... 343, 348
 Grapher 199
graphicx 87, 348
 GRASS 34
 GTK+ 338
 GUI for PDFTK 37
 Gurari, Eitan 39
gzip 299

H

Hagen, Hans 37
 Haralambous, Yannis ... 332
 Harders, Harald 122
hcentering 244
hdivide 243
head 244
headheight 244
headline 45
headsep 244
height 242
heightrounded 243
 Heinz, Carsten 284, 350
 help 12, 13
 helvet 328, 330
 Helvetica 330
 hline 87
 his 337
hmargin 244
hmarginratio 244
 Hobby, John 213
hoffset 245
howpublished (B \TeX)
 230, 233
hscale 242

ht 40
 htm 42
 HTML i, 4, 28
 html 41
 HTML タグ 43
 Hyna, Irene 346
 Hyper Link 28
 hyperref . vi, 42, 44, 292, 350
 HyperTeX 30

I

i18n 34
 identify 189
 .idx (拡張子) 22
 IEEE 337
 ieice 337
 ieicej 337
 ifthen 309
 ignoreall 243
 ignorefoot 243
 ignorehead 243
 ignoreheadfoot 243
 ignoremp 243
 .ilg (拡張子) 22
 Illustrator 198, 204
 ImageMagick 39
 inbook (文献の種類) 229
 includeall 243
 includefoot 242
 includehead 242
 includeheadfoot 242
 includemp 242
 incollection (文献の種類)
 229
 .ind (拡張子) 22
 indent 54
 indentation 45, 53
 indentfirst 54, 88, 322
 info 13
 inner 243
 inproceedings (文献の種
 類) 229, 233
 INRIA 338
 .ins (拡張子) 21
 International Phonetic
 Alphabet 337
 intlimits 151
 IPA
 — フォント 34
 ISO9660 42
 .ist (拡張子) 22

J

j-article 41

j-book 41
 j-report 41
 jabbrv 230
 JabRef 236
 jalpha 230
 Japanese 32
 jarticle .. 24, 48, 84, 87, 313,
 333
 jarticle.cls 19
 jBibTeX 219, 222
 jbook 26, 84, 237, 333
 jclasses 85
 Jeffrey, Alan v, 347
 Jeichter, Jerry 180
 Jensen, Frank 307, 349
 JIS X 0208 91, 332
 JIS 基本漢字 332
 jlshort 346
 jmakeindex 44
 journal (BibTeX) . 230, 232
 JPEG 189
 jpg 41
 .jpg (拡張子) 23, 30
 jplain 230
 jplain.bst 225, 226
 jpsj2 337
 jreport 84, 88, 318, 333
 jsarticle 86, 333
 jsbook 52, 86, 237, 241, 245,
 318, 320, 333
 jsclasses .. 5, 31, 85, 86, 241,
 255, 275, 333
 jsize10.clo 26
 jspf 86
 junrst 230

K

Kelley, Colin 217, 338
 kelvin 57
 Kennedy, Bill 341
 Kernighan, Brian 212
 kerning 69
 key (BibTeX) 230
 Keynotes 199
 keywordstyle 289
 Kiddle, Oliver 339
 kilo 57
 kilogram 57
 Kingdon, Jim 307
 kinsoku.tex 26
 Knappen, Jörg 337, 348
 Knuth, Donald . v, 3, 10, 81,
 213, 327, 331, 344
 Koch, Richard 27

Korean 32
 Kpathsearch 19, 38
 kpsewhich 19
 Kwok, Conrad 209

L

label 289
 Label ‘key’ multiply
 defined 112
 Label(s) may have
 changed. Return
 to get
 cross-ferences
 right. 112
 labelfig 201
 Lambda 332
 Lamed 332
 Lamport, Leslie iv, v, 4,
 342, 347
 landscape 85, 242
 language 288
 Larsson, Alexander 338
 Larsson, Jan-Ake 338
 lastline 288
 L^AT_EX 4
 L^AT_EX
 — 原稿 11
 — 処理状況 11
 — での計算 ... 87, 307
 — とワープロの違い . 53
 — における単位 56
 — の動かし方 7
 — のエラー 14
 — の基本 7
 — の旧版 29
 — の原稿執筆支援 . 12
 — の最低限の規則 .. 1
 — の実行 8
 — の中途ファイル ... 11
 — のログファイル ... 11
 L^AT_EX 3 5, 327
 L^AT_EX 2.09 5
 L^AT_EX 2_ε 5
 L^AT_EX2HTML 338
 latex2html 294
 L^AT_EX2RTF 338
 latexmk 225, 306
 latexsym 136
 layout 88, 237
 leading 70
 left 243
 leftidx 122
 legalpaper 241
 Lehman, Philipp 350

- Lemberg, Werner 349
length 57
leqno 85, 151
less 13, 19
letter 46
——space 70
letterpaper 85, 241
ligature 69
line break 68
lines 242
lineskip 288
linewidth 288
List of Figures 50
List of Tables 50
listfile.tex 26
Listings 291
listings 284, 287
lmargin 243
lmodern 328, 330
lmodern.sty 328
.lof (拡張子) 22
.log (拡張子) 22
longtable 87, 181, 183
.lot (拡張子) 22
ls 13
ls-R 336
lshort 346
ltxboxes.dtx 265
ltxplain.dtx 277
luminous intensity 57
Lunde, Ken 341
LyX 338
- M**
- Mac OS X 327
——でのプレビュー 29
——の ATSUI 327
——の執筆支援環境 27
——への導入 5
MacOS X WorkShop 5
MacWiki 5
Make 27, 225, 296, 306
make 340
makebb 195
Makefile 27, 298
MakeIndex 22
man 13
manual (文献の種類) 229
Map ファイル 33
Marché, Claude 338
margin 244
marginpar 245
marginparsep 243, 245
marginparwidth 243, 245
- marginratio* 244
mass 57
masterthesis (文献の種類)
229
Mathematica 199
mathpazo 81, 328
mathptmx 81, 328
MATLAB 198, 199, 341
McPherson, Kent 347
Meadow 340
mega 57
mendex 22, 44
METAFONT 198, 213
METAPOST 23, 198, 213
meter 57
micro 57
Microsoft Office 3
milli 57
misc (文献の種類) 229
Misplaced alignment tab
character & 14
Missing \$ inserted 15
Mittelbach, Frank . iv, v, 82,
136, 138, 150, 260,
263, 342, 347
mkdir 12, 13
mktexlsr 35, 336
mole 57
month (BIBTEX) 230, 233
Moore, Ross 217
Morgan, Tim 212
Morisawa 333
move 12
.mp (拡張子) 23
mpfootnote (カウンタ) 106
multicol .. 88, 260, 261, 263,
264
multitrow 180
Musciano, Chuck 341
Musliner, David J. 348
mv 13
Mxdvi 29
- N**
- namelimits* 151
nano 57
nest 73
nested sections 48
newcent 328
Newham, Cameron 339
next 42
Ngraph 338
Niepraschk, Rolf 211
nofoot 244
- nohead* 244
noheadfoot 244
nointlimits 151
nomarginpar 245
nonamelimits 151
nosort 235
nospace 235
nosumlimits 151
note (BIBTEX) 223, 230
note 45, 59
notitlepage 85
number (BIBTEX) .. 230, 232
numbers 289
numberstyle 289
- O**
- Oberdiek, Heiko ... 194, 348
obsolete 26
Octave ... 198, 217, 338, 341
Oetiker, Tobias ... 346, 348
offset 245
okumacro 66, 75, 333
okuverb 333
Omega 332
OmniGraffle 199
onecolumn 85
oneside 85, 239, 244
Oostrum, Piet 180, 250, 350
openany 85
OpenOffice.org 3, 339
openright 85
OpenType 352
Oram, Andrew 340
organization (BIBTEX) 230
Orwant, Jon 340
OTF 91, 332, 335
outer 243
Oxford Style 345
- P**
- package 82
page (カウンタ) .. 106, 107
Pages 199
pages (BIBTEX) 230, 232
Pakin, Scott 348, 349
Palatino 328, 331
paperheight 241
papersize 31, 86, 241
paperwidth 241
paragraph (カウンタ) . 106
paragraph 45, 46, 49
——skip 70
part (カウンタ) 106
part 46, 49

- Partl, Hubert 346
 Patashnik, Oren v, 219, 349
 PATH 194
 PDF 30, 189
 — 画像の張り込み . 188
 — でのフォントの種類 . . . 36
 — でのフォント名 . . . 36
 — と Mac OS X . . . 199
 — のアクセシビリティ . 37
 — の圧縮率 31
 — の暗号化 32
 — の解像度 31
 — の互換性 31
 — のテキストのコピー . . . 36
 — のバージョン 28
 — のパスワード 32
 — のプレビュー 30
 — の編集 199
 — の編集作業 36
 — の文字列検索 36
 — の文字列の抽出 36
 — の用紙サイズ 37
 — ファイル変換時のエラー 32
 — ブックマーク 30
 PDF
 — のセキュリティ 32
 pdfcrop 194
 pdf \LaTeX 327
 pdf \TeX 327
 pdffonts 35
 pdfimages 35
 pdfinfo 35, 36, 194, 296
 pdf \LaTeX 28, 245, 327
 pdf \TeX 245, 327
 pdftex 245, 294
 PDFtk 37, 338
 pdftops 35, 37
 pdftotext 35, 37
 pdftricks 216
 pdvips 38
 Peek, Jerry 339
 Perl 186, 196, 306, 340
 phdthesis (文献の種類) . . . 229
 Phoenix, Tom 340
 Photoshop 198
 PIC 212
 pic-align 41
 pic-array 41
 pic-eqnarray 41, 42
 pic-equation 41
 pic-m 41
 pic-m+ 41
 pic-matrix 41
 pico 57
 pict2e 186, 211
 Pic \TeX 198
 Plaice, John 332
 plain 230
 plain \TeX 348
 plain2 203
 p \LaTeX 5
 pldefs.ltx 26
 PNG 189
 png 41
 .png (拡張子) 23, 30
 Podar, Sunil 208, 209
 Polk, Jeff 307
 portrait 242
 postbreak 288
 PostScript 38
 PPM 188
 prebreak 288
 PrimoPDF 197
 proc 84
 proc.dtx 84
 proceeding 233
 .ps (拡張子) 28
 ps2jpdf 194, 338
 ps2pdf 294
 ps2pdf 194
 pst-all 215
 pst-col 215
 pst2pdf 216
 PSTricks 198, 215
 pstricks 350
 psutils 39
 p \TeX 5
 publisher (Bib \TeX) . . . 223, 230, 232
 punctuation 45, 57
 pxdvi 17
 pxfonts . . . 81, 142, 165, 257, 328, 330, 331, 337
Q
 QPL 199
R
 R 198, 338
 Radhakrishnan, CV . . . 216
 Rahtz, Sebastian . iv, v, 292, 342
 Raichle, Bernd 349
 Rajagopal, CV 216
 Raymond, Eric 340
 Reckdahl, Keith . . . 348, 350
 Red Hat 17, 38
 Ref for Windows 236
 Reference ‘key’ on page
 n undefined . . 112
 references 219
 report 84
 report 86, 333
 report.cls 84
 reversemarginpar 245
 reversemp 245
 right 243
 Ring Server 347
 Ritter, Robert 345
 rm 13
 rmargin 243
 rmlv 34
 rml 34
 Robbins, Arnold 340
 Rokicki, Tomas 38
 Rose, Kristoffer 217
 Rosenblatt, Bill . . . 339, 340
 Rowley, Chris . . v, 347, 349
 Rplots.pdf 198
 Ruby 341
 rulecolor 290
 rulesepcolor 290
 Ryu, Young . . . 81, 165, 328
 Ryumin-Light 34
S
 Séroul, Raymond 201
 Samarin, Alexander . . . 342
 scale 242
 scaled 330
 Schöpf, Rainer . . v, 82, 150, 347, 349
 Schlegl, Elisabeth 346
 Schmidt, Walter 350
 school (Bib \TeX) 230
 Schwartz, Randal L. . . . 340
 SciLab 217
 screen 241
 secnumdepth (カウンタ) 51
 second 57
 section (カウンタ) . . . 106
 section 46, 49
 section+ 42
 sectioning 48
 sed 196
 sentence 46
 — space 70
 serial number 48

- series (BIBTEX) .. 230, 232
 setspace 55
 showframe 245
 showkeys 88, 112
 showspace 288
 showtabs 288
 SI 56
 Siebenmann, Laurent ... 201
 size10.clo 84
 size11.clo 84
 size12.clo 84
 slides 84
 slides.dtx 84
 Smith, Brian 338
 space 235
 space 70
 letter— 70
 sentence— 70
 word— 70
 src-special 337
 Stallman, Richard iv, vi
 Stephenson, Peter 339
 stepnumber 289
 Steward, Sid ... 37, 338, 341
 Strang, John 339
 stringstyle 289
 .sty (拡張子) 21
 subfigure 200
 subparagraph (カウンタ) ..
 106
 subparagraph 49
 subsection (カウンタ) 106
 subsection 49
 subsubsection (カウンタ) ..
 106
 subsubsection 49
 sunlimits 151
 Susie 187
 Synaptic 5, 336
 syntonly 82
- T**
- T1 60
 t4ht 44
 tab 288
 table (カウンタ) 106
 table 172
 tabsize 288
 tabularx 87, 184, 185
 Talbott, Steve 340
 tarticle 84
 tbook 84
 tbtags 151
 TDS 348
- tera 57
 test.tex 9
 teTEX 337
 TEX 3
 .tex (拡張子) 22
 tex4ht 40
 tex4ht 294
 texdoc 337
 texexec 37
 texhash 336
 Texinfo 284
 TEXmacs 338
 texmf.cnf 19
 TEXShop 27
 text 242
 textcomp 57, 256, 330
 textheight 242, 245
 textwidth 242, 245
 Tgif 26, 198, 338, 342
 Thành, Hàn Thế 327
 theorem 88, 138
 Thorup, Kresten 349
 Thorup, Kresten Krab .. 307
 time 57
 Times 320, 328, 330
 tips ii
 title (BIBTEX) ... 223, 230
 title 45, 46
 titlepage 85
 tmargin 243
 .toc (拡張子) 22
 tocdepth (カウンタ) 51
 Todino, Grace 339
 tools 87
 top 243
 total 242
 totalheight 242
 totalwidth 242
 TPIC 212
 treport 84
 Trinkle, Daniel 338
 TrueType 34, 36
 TtH 39
 TUG 348
 twocolumn 85, 245
 twoside ... 85, 239, 243–245,
 252
 txfonts ... 81, 142, 165, 257,
 328, 330, 331, 337
 Type1 36
 type1cm 328, 330, 332
 type1ec 328, 330, 337
 Type3 36
 typeface 76
- typical 178
- U**
- Unable to open <file>.pdf .
 32
 underline 59
 Unicode 32
 Unix 系 OS 29, 196
 —でのプレビュー ... 29
 unsrt 230
 url 44, 233, 292
 usenames 282
 UTF 332
- V**
- varoref 88
 vcentering 244
 vdivide 243
 verbatim 88, 337
 verbatim 62
 verbose 245
 Vesperman, Jennifer 340
 Vine Linux 26, 339
 —への導入 5
 vmargin 244
 vmarginratio 244
 voffset 245
 Vojta, Paul 337
 Volovich, Vlandimir 349
 volume (BIBTEX) .. 230, 232
 vscale 242
- W**
- Wall, Larry 340
 warning 24
 white space 60
 Wicks, Mark v, 30, 349
 width 242
 Williams, Graham 346
 Williams, Thomas . 217, 338
 Windows 8
 —でのタイプセット .. 9
 —でのファイル操作 .. 12
 —でのプレビュー ... 29
 —の執筆支援環境 .. 27
 —のプレビューア ... 17
 —への導入 5
 Wolinski, Marcin 349
 Wooding, Mark 63, 349
 Word 3, 328
 word 46
 —space 70
 Writer 3
 WWW 233

WYSIWYG 3, 337

X

X11 5
 xdvi 17, 337
 xdvi 337
 Xe_{La}TeX 327
 Xfig 198
 xhtml 41
 xleftmargin 288
 XML i
 xml2bib 338
 Xpdf 30, 35
 xr 88
 xrightmargin 288
 XSL i
 xspace 88, 281
 Xy-pic 217
 Xe_{La}TeX 217

Y

YaTeX 26, 296, 337
 year (BiBTeX) 223, 230
 yomi (BiBTeX) 223, 232
 You can't use 'macro
 parameter
 character #' .. 14
 You can't use
 '\spacefactor' in
 vertical mode 100
 You have requested, on
 input line n .. 24

Z

Zandt, Timothy ... 215, 270,
 273, 350
 Zuhn, David 307

あ

空き 54, 68, 274
 垂直方向の— 275
 水平方向の— 274
 単語間の— 69
 文間の— 69
 文字間の— 68
 秋田純一 v
 アク्यूト 61
 アクセシビリティ 37
 アクセント 60
 大きい— 135
 数式中の— 135
 小さい— 135
 文中の— 61
 アクセント記号 ... 60, 61, 328

アクティブ文字 99
 圧縮率 31
 アットマーク 98, 223
 阿部昌平 185
 阿部紀行 5
 アラビア数字 149, 251
 暗号化 33
 アンパサンド 12, 116, 126, 174
 アンペア 57

い

生田誠三 342
 いけだやすし 339
 石坂智 338
 石塚圭樹 341
 意思伝達手段 76
 イタリック体 79
 イタリック補正 60
 一意 48, 171
 市川順一 339
 一括処理 4
 伊藤敏 342
 稲垣徹 352
 イニシャルコマンド 19
 井上祥史 342
 入れ子 73
 岩熊哲夫 346
 印刷 229
 —できるファイル ... 11
 両面— 240
 印刷所 33, 38
 印刷面 85
 インチ 56
 引用 20, 54, 58, 219
 —の引用 64
 雑誌名の— 65
 書籍名の— 65
 単語の— 63
 段落の— 63
 複数段落の— 64
 文の— 54, 63, 64
 本の名前の— 65
 引用符 20, 21, 65, 70
 欧文の— 21
 和文の— 21

う

ウェブ 61
 ウェブブラウザ 28
 ウェブページ 233
 上余白 243
 受け 58
 白田昭司 342
 内田昭宏 203

内山孝憲 29, 351
 ウムラウト 61
 梅木秀雄 240
 浦壁厚郎 185
 上付き 121
 上付き文字 99

え

英字 149
 英文字 99
 エクスプローラ 189
 江口庄英 343
 エスケープ文字 99
 閲覧日 233
 エラー 14
 Label 'key' multiply
 defined 112
 Label(s) may have
 changed. Return
 to get
 cross-ferences
 right. 112
 Misplaced alignment
 tab character & .
 14
 Missing \$ inserted ..
 15
 Reference 'key' on
 page n undefined
 112
 Unable to open
 <file>.pdf 32
 You can't use
 '\spacefactor' in
 vertical mode 100
 You can't use 'macro
 parameter
 character #' .. 14
 エラーメッセージ 10
 円 206
 円記号 20
 円弧 210
 エンコーディングマップ ... 34

お

欧文
 —の引用符 21
 大石勝 351
 大型演算子 134
 大澤英一 232
 大島利雄 v, 17, 29, 348, 351
 オートマトン 217
 大友康寛 5, 27, 352
 大野修一 217

- 尾田晃 217
 奥村晴彦 . v, 52, 81, 85, 232,
 284, 333, 335, 342,
 351
 奥村本 342
 起こし 58
 オゴネク 61
 小田忠雄 148
 乙部蔵己 v, 343, 351
 オフィスソフト 3
 オプション
 10pt 56, 85, 86
 11pt 85, 86
 12pt 85, 86
 12Q 86
 14pt 86
 14Q 86
 17pt 86
 20pt 86
 21pt 86
 25pt 86
 30pt 86
 36pt 86
 43pt 86
 9pt 86
 a0paper 241
 a1paper 241
 a2paper 241
 a3paper 241
 a4j 85, 86
 a4paper 85, 86, 241, 294
 a4var 86
 a5j 85, 86
 a5paper 85, 86, 241, 294
 a6paper 86, 241
 asymmetric 244
 b0paper 241
 b1paper 241
 b2paper 241
 b3paper 241
 b4j 85, 86
 b4paper 85, 86, 241
 b5j 85, 86
 b5paper 85, 86, 241, 294
 b5var 86
 b6paper 241
 bindingoffset 244
 bmargin 243
 body 242
 bookmarks 294
 bookmarksnumbered ...
 294
 bookmarksopen 294
 bookmarkstyle 294
 bookmarkstyleopenlevel ..
 294
 bottom 243
 breaklinks 294
 centering 244
 centertags 151
 ceqno 151
 charset 41
 colorlinks 294
 columnsep 245
 divide 243
 draft 85, 202
 dvipdfm .. 187, 245, 294
 dvipdfmx 187, 188, 196
 dvips 188, 196, 245, 294
 dvipsnames 282
 english 86, 333
 executivepaper 241
 expert 335
 final 85, 202
 fleqn 85, 115, 151
 fn-in 41
 font+ 41
 foot 244
 footskip 244
 gif 41
 hcentering 244
 hdivide 243
 head 244
 headheight 244
 headsep 244
 height 242
 heightrounded 243
 hmargin 244
 hmarginratio 244
 hoffset 245
 hscale 242
 htm 42
 html 41
 ignoreall 243
 ignorefoot 243
 ignorehead 243
 ignoreheadfoot 243
 ignoremp 243
 includeall 243
 includefoot 242
 includehead 242
 includeheadfoot 242
 includemp 242
 inner 243
 intlimits 151
 jpg 41
 landscape 85, 242
 latex2html 294
 left 243
 legalpaper 241
 leqno 85, 151
 letterpaper 85, 241
 lines 242
 lmargin 243
 margin 244
 marginpar 245
 marginparsep . 243, 245
 marginparwidth ... 243,
 245
 marginratio 244
 namelimits 151
 next 42
 nofoot 244
 nohead 244
 noheadfoot 244
 nointlimits 151
 nomarginpar 245
 nonamelimits 151
 nosort 235
 nospace 235
 nosumlimits 151
 notitlepage 85
 offset 245
 onecolumn 85
 oneside 85, 239, 244
 openany 85
 openright 85
 outer 243
 paperheight 241
 papersize ... 31, 86, 241
 paperwidth 241
 pdftex 245, 294
 pic-align 41
 pic-array 41
 pic-equarray 41, 42
 pic-equation 41
 pic-m 41
 pic-m+ 41
 pic-matrix 41
 png 41
 portrait 242
 ps2pdf 294
 report 86, 333
 reversemarginpar .. 245
 reversemp 245
 right 243
 rmargin 243
 scale 242
 scaled 330
 screen 241
 section+ 42
 showframe 245

space 235
 sumlimits 151
 T1 60
 tbtags 151
 tex4ht 294
 text 242
 textheight 242, 245
 textwidth 242, 245
 titlepage 85
 tmargin 243
 top 243
 total 242
 totalheight 242
 totalwidth 242
 twocolumn 85, 245
 twoside 85, 239,
 243–245, 252
 usernames 282
 vcentering 244
 vdivide 243
 verbose 245
 vmargin 244
 vmarginratio 244
 voffset 245
 vscale 242
 width 242
 xhtml 41
 ドキュメントクラス—— 31
 オフセット 31, 245
 垂直方向の—— 31
 水平方向の—— 31
 乙部本 351
 オミクロン 132
 親カウンタ 138
 親ディレクトリ 14
 折れ線 209
 音声記号 337

か

会議 230, 233
 改行 68
 ——された表示 15
 ——の抑制 70
 ——を許さない空き 274
 ——を許す空き 274
 array 環境での—— 127
 eqnarray 環境での——
 116
 L^AT_EX での—— 60
 行揃えにおける—— 71
 原稿における—— 53
 出力結果における—— 53
 数式モード中の—— 113
 ——の位置 67

——の禁止 68
 文章中の—— 68
 連続した—— 61
 行末文字としての—— 99
 会議録中の論文 229
 解像度 31
 書いたまま出力する 62
 改段落 68
 改丁 254
 回転 192
 表の—— 193
 文字列の—— 192
 概要 52
 回路図 215, 217
 会話文 20
 カウンタ 105
 ——の新設 108
 ——の設定 108
 chapter 106
 enumi 106
 enumii 106
 enumiii 106
 enumiv 106
 equation 106, 159
 figure 106
 footnote 106
 mpfootnote 106
 page 106, 107
 paragraph 106
 part 106
 secnumdepth 51
 section 106
 subparagraph 106
 subsection 106
 subsubsection 106
 table 106
 tocdepth 51
 親—— 138

化学

——構造式 217, 344
 ——式 217, 344
 可換図 159
 かぎ括弧 .. 20, 21, 58, 63, 65
 書き出し 197
 学位論文 45
 角括弧 20, 21, 145
 ——で囲まれた引数 .. 20
 ——の数式での出力 145
 拡大 192
 区切り記号の—— 126
 図の—— 190
 ページの—— 30
 文字列の—— 192
 拡張子 21, 83

——の省略 32
 .aux 10, 22, 50, 225
 .bb 23
 .bb1 22, 224
 .bib 22
 .blg 22
 .bmp 23, 30
 .bst 22
 .clo 22
 .cls 21
 .dtx 21
 .dvi 22, 28
 .epdf 30
 .eps 22, 30
 .fd 22
 .idx 22
 .ilg 22
 .ind 22
 .ins 21
 .ist 22
 .jpg 23, 30
 .lof 22
 .log 22
 .lot 22
 .mp 23
 .png 23, 30
 .ps 28
 .sty 21
 .tex 22
 .toc 22
 角藤亮 v, 215, 294, 348, 351
 簡条書き 73
 記号付きの—— 73
 説明付きの—— 73
 番号付きの—— 73
 数値の代入 56
 下線 59, 269
 ——アクセント 61
 ——の意味 59
 画像
 ——の回転 190
 ——の拡大 190
 ——の切り抜き 190
 ——のクリッピング .. 191
 ——の原点座標 189
 ——のサイズ情報 30
 ——の高さ 190
 ——の縦横比 191
 ——のトリミング 190
 ——のバウンディングボックス 190
 ——の幅 190
 ——の張り込み 186
 ——ベクトル 22

- への文字の追加 201
 —編集 26
 可逆圧縮— 22
 単一ページの— 28
 浮動体としての— 172
 フルカラー— 22
 無圧縮— 22
 仮想プリンタ 197
 画像編集 338
 片面印刷 244
 学会 337
 IEEE 337
 情報処理— 337
 人工知能— 337
 ソフトウェア科— 337
 電子情報通信— 337
 日本物理— 337
 認知科— 337
 ヒューマンインタフェース
 — 337
 学会誌 232
 括弧
 —のある行列 87
 —の大きさの調整 125
 —の足りない状態 17
 引用のための— 65
 書籍名のための— 65
 数式モード中の— 114
 場合分けの— 129
 合字 69
 AEの— 61
 OEの— 61
 カテゴリーコード 98
 下点アクセント 61
 かな漢字変換プログラム 9
 可変長の長さ 263
 カラースペシャル 30
 カレントディレクトリ 14
 川原稔 341
 巻 232
 環境 20, 92
 abstract 52
 align 153, 155, 162
 alignat 153, 162
 aligned 162
 alignedat 162
 alltt 63
 appendix 278
 array 87, 126, 153, 173,
 180, 184
 Bcenter 271
 Bdescription 272
 Benumerate 272
 Beqnarray 271
 Bflushleft 271
 Bflushright 271
 Bitemize 272
 Bmatrix 156
 bmatrix 156
 cases 145, 153
 CD 150
 cempth 94
 center 71, 72
 comment 61
 dashjoin 208
 description 73
 displaymath 115
 document 19
 dottedjoin 208
 drawjoin 208
 enumerate ... 73, 74, 87
 eqnarray 109, 117
 eqnarray* 116
 equation 115
 figure .. 172, 173, 261,
 263
 figure* .. 260, 261, 263
 filecontents 23
 flalign 153, 155
 flushleft 71, 72
 flushright 71, 72
 framebox 273
 fverbatim 272
 gather ... 152, 154, 162
 gathered 162
 indentation 54
 itemize 73
 longtable 181
 lrbox 268
 lstlisting 285
 math 114
 matrix .. 128, 153, 156,
 157
 minipage 200, 267
 multicols 261
 multline 153
 myfigure 261, 263
 mytable 261, 263
 picture . 172, 186, 205,
 211
 pmatrix 128, 156
 quotation 64
 quote 54, 64
 Sbox 270
 spacing 55
 split 153, 154
 subarray 157, 158
 subequations 158
 tabbing 173
 table ... 172, 173, 181,
 261, 263
 table* ... 260, 261, 263
 tabular .. 87, 172, 173,
 176, 179, 180, 184
 tabular* 87
 tabularx 184, 185
 thebibliography .. 22,
 220, 257
 theindex 22
 Verbatim 273
 verbatim .. 62, 88, 203,
 272, 273, 333
 Vmatrix 156
 vmatrix 156
 定理型の— 138
 問題型の— 138
 例題型の— 138
 関係子 134
 韓国語 32
 間違いの修正 81
 カンデラ 57
 巻末 219, 224
- ## き
- キートップ iii
 キービット 32
 キーボード iii
 —からの入力 ii
 ギガ 57
 機関名 230
 記号 90, 257
 —の意味 2
 —の重ね合わせ 141
 —の前の空白 2
 —の使い方 2, 3
 —の積み重ね 141
 —の分類 90
 アクセント— 61
 円— 20
 大型演算子 134
 大きさが可変の— 123
 音声— 337
 関係子 134
 区切り— 124, 169
 校正— 81
 根号— 143
 演算子 134
 数学— . 113, 132, 165,
 169
 積分— 123
 添え字における— .. 124
 単位— 56

通貨—— 257
 特殊な—— 60, 61, 92
 二項演算子 134
 文を区切る—— 45
 偏微分—— 140
 矢印 135
 横棒の—— 66
 ルート—— 143
 木構造 215
 記事の検索 48
 機種依存文字 91
 奇数起こし 254
 木下是雄 221
 基本書体 81
 木村健一 v
 疑問符 14, 16
 脚注 59, 79, 292
 2 段組での—— 87
 minipage 環境での—— 267
 表中の—— 177
 行
 ——分割 150
 行送り 55, 287, 321
 行数 242, 247
 行間 70
 行間空白 69
 強調 3, 59, 78
 ——の中の単位 57
 見出しの—— 80
 文字列の—— 59
 和文の—— 59, 80
 行頭
 ——の字下げ 54
 行末 274
 ——のコンマ 223
 ——文字 99
 行列
 ——中の小数点 87
 ——に付ける括弧 87
 ——の罫線 87
 ——の幅 127
 括弧付きの—— 128
 行列演算 338
 許可 32
 局所変数 101
 曲線
 ——の描画 206
 2 次ベジェ—— 211
 3 次ベジェ—— 211
 Chaikin—— 210
 近似—— 204
 スプライン—— 204
 ベジェ—— 204

ギリシャ文字 132, 142
 ——の大文字 132
 ——の小文字 132
 ——の変体大文字 160
 ——の変体小文字 133
 キリル文字 349
 キロ 57
 キログラム 57
 近似曲線 204
 禁則処理 81

く

空行 61, 113
 空白 99
 ——の調整 119
 数値と単位の—— 57
 行間の—— 69
 原稿中の—— 60
 コマンドの後の—— 281
 四分空き 70
 数式モード中の—— 113
 正しい量の—— 68
 単語間—— 274
 段落間の—— 69
 著者名中の—— 223
 区切り 124
 ——記号 124
 項目の—— 235
 コマンドの—— 97
 単語の—— 66
 著者の—— 47
 文の—— 63
 ページの—— 253
 区切り記号 124
 くさび 61
 句読点 45, 57
 熊澤吉紀 352
 組版 1, 2
 クラス 21, 82
 ——ファイル 21
 article 48, 84, 313
 book 84
 classes 84
 ieice 337
 ieicej 337
 j-article 41
 j-book 41
 j-report 41
 jarticle 24, 48, 84, 87,
 313
 jbook 26, 84, 237
 jclassess 85
 jpsj2 337
 jreport 84, 88, 318

jsarticle 86
 jsbook 52, 86, 237, 241,
 245, 318, 320
 jsclasses 85, 86, 255, 275
 jspif 86
 proc 84
 report 84
 slides 84
 tarticle 84
 tbook 84
 treport 84
 小規模な文書用の—— 84
 書籍用の—— 84
 標準的な—— 83
 報告書用の—— 84
 クラスオプション 85
 グラデーション 215
 グラフ描画 338
 グリフ 36
 グルー 276
 グルーピング 92, 113, 114
 グループ
 ——の終わり 99
 ——の開始 99
 グレーブ 61
 グローバルオプション 25
 グローバル化 102

け

警告 16, 24
 You have requested,
 on input line n .
 24
 煩雑な—— 16
 計算機科学者 327
 形式言語 90
 罫線 269
 ——の太さ 266
 2 段組での—— 239
 行列の—— 127
 表中の—— 174
 表の—— 172
 フッタ上部の—— 251
 ヘッダ上部の—— 251
 ケルビン 57
 原稿 7
 ——から PDF の作成 28
 ——作成時の注意 11
 ——作成の支援 26, 296
 ——中の改行 53
 ——中の空白 60
 ——の校正 81
 ——の構成 19
 ——の再コンパイル 27

—の出力形式 27
 —の書式 21
 —の処理 4
 —の先頭 23, 29
 —の先頭部分 19
 —の入力支援 12
 —の版管理 306
 —のブリアンブル 23
 —の分割 279
 —の編集 7
 —の文字サイズ 85
 —の用紙サイズ 85
 執筆段階の— 85
 ドラフト段階の— 85
 原点座標 189

ㇿ

弧 210
 校正 81
 合線 61
 光度 57
 後方参照 227
 コード
 カテゴリー— 98
 分類— 98
 国際音標文字 337
 国際単位 56
 小口 243
 黒板風書体 118
 ゴシック体 34, 80
 小関吉則 340
 固定長の長さ 263
 コマンド 7, 8, 12, 20
 —の後の空白文字 88
 —の情報表示 13
 —の引数 20
 —のヘルプの表示 12
 —ラインオプション 13,
 38
 イニシャル— 19
 宣言型の— 103
 内部— 13
 ブリアンブル— 19
 命令型の— 103
 コマンドプロンプト iii, 8, 12
 込め物 276
 コメント 61
 —アウト 61
 —文字 99
 コロン 58
 コンソール iii, 8
 コントロール
 —シークエンス 91
 —シンボル 91

—スペース 91
 —ワード 91
 コンパイル 8
 —の回数 ii
 コンマ 25, 58
 全角の— 57

さ

サーカムフレックス 61
 サイズ 79
 彩度 281
 齋藤修三郎 91, 332, 337, 352
 坂上貴之 341
 索引
 —の見出し 259
 作者 46
 作図 186
 作文 2
 —技術 221
 さざなみフォント 34
 雑誌
 和文の— 65
 雑誌名
 —の引用 65
 佐藤竜一 339
 サブセット 36
 左右
 —対称 244
 —非対称 244
 左右起し 85
 参考文献 219, 259
 —データベース 220
 —の補足事項 221
 —の見出し 259
 三次元射影 215
 サンセリフ体 79

し

時 254
 シェイブ 79
 イタリック— 79
 スモールキャピタル— 79
 スラント— 79
 シェル iii, 8, 12
 字間 68
 時間 57
 色相 281
 字形 36
 字下げ 45, 53
 —の幅の調節 54
 —の抑制 54
 行頭の— 54
 段落の— 54
 段落始めの— 53

箱の中での— 267
 見出し直後の— 88
 自然言語 90
 下付き 121
 —文字 99
 下余白 243
 執筆環境 26
 字詰め 69
 質量 57
 四分空き 70, 81
 嶋田隆司 343
 紙面
 —の拡大縮小 56
 シャープ S 61
 斜線 178
 斜体 3
 住所 230
 修飾子 56, 74
 修士論文 229
 縮小 192
 出版社 33, 38, 221, 229, 230,
 232
 出版年 232
 演算子 134
 —の否定 141
 大型— 134
 二項— 122, 134
 章 46, 259
 —の見出し 259
 上位互換 32
 小括弧 21
 商業出版 3
 乗算 154
 小数点 87, 179
 状態遷移図 217
 章立て 52, 172, 318
 小なり 12
 章標 61
 情報 45
 情報工学 336
 情報処理学会 337
 情報処理推進機構 34
 商用ディストリビューション 34
 ショート 61
 色 281
 ページの— 283
 文字の— 283
 書式 i, 21, 82
 —と内容の分離 i
 原稿の— 21
 書籍
 和文の— 65
 書籍名
 —の引用 65

- 書体 ii, 76
 ——の種類 ii
 ——の属性 22
 助動詞 96
 書名 20, 221, 232
 処理
 ——の中断 16
 シリーズ 79, 221, 232
 ボールド—— 79
 ミディアム—— 79
 資料の配布 30
 白黒印刷 30
 シングルクオート 21, 63
 人工知能学会 337
 人名
 Abrahams, Paul W. 348
 Alver, Morten 236
 Arseneau, Donald . 234, 292
 Batada, Nizar 236
 Beebe, Nelson ... v, 338
 Berliner, Brian 307
 Berners-Lee, Tim ... 89
 Berry, Karl 38, 350
 Braams, Johannes v, 62, 347
 Calishain, Tara 341
 Cameron, Debra ... 340
 Carlisle, David ... v, 74, 112, 129, 179, 181, 184, 282, 309, 347
 Chen, Pehong v
 Cheng, William
 Chia-Wei . 199, 338
 Christiansen, Tom . 340
 Cochran, Steven Douglas 200
 Collins, John 306
 Doerr, Hans-Peter . 186
 Doob, Michael 348
 Dornfest, Rael 341
 Dougherty, Dale ... 340
 Downes, Michael v
 Drakos, Nikos 338
 DuBois, Paul 339
 Duchier, Denys 349
 Duggan, Angus 39
 Eaton, John ... 217, 338
 Eijkhout, Victor 61
 Esser, Thomas 337
 Fear, Simon 178
 Filliâtre,
 Jean-Christophe ... 338
 Fogel, Karl 340
 Friedl, Jeffrey E. F. 340
 Goossens, Michel iv, 342
 Grätzer, George ... 343, 348
 Gurari, Eitan 39
 Hagen, Hans 37
 Haralambous, Yannis ... 332
 Harders, Harald 122
 Heinz, Carsten 284, 350
 Hobby, John 213
 Hyna, Irene 346
 Jeffrey, Alan ... v, 347
 Jeichter, Jerry 180
 Jensen, Frank . 307, 349
 Kelley, Colin .. 217, 338
 Kennedy, Bill 341
 Kernighan, Brian .. 212
 Kiddle, Oliver 339
 Kingdon, Jim 307
 Knappen, Jörg 337, 348
 Knuth, Donald v, 3, 10, 81, 213, 327, 331, 344
 Koch, Richard 27
 Kwok, Conrad 209
 Lamport, Leslie iv, v, 4, 342, 347
 Larsson, Alexander 338
 Larsson, Jan-Ake .. 338
 Lehman, Philipp ... 350
 Lemberg, Werner .. 349
 Lunde, Ken 341
 Marché, Claude 338
 McPherson, Kent .. 347
 Mittelbach, Frank iv, v, 82, 136, 138, 150, 260, 263, 342, 347
 Moore, Ross 217
 Morgan, Tim 212
 Musciano, Chuck .. 341
 Musliner, David J. . 348
 Newham, Cameron . 339
 Niepraschk, Rolf ... 211
 Oberdiek, Heiko ... 194, 348
 Oetiker, Tobias 346, 348
 Oostrum, Piet 180, 250, 350
 Oram, Andrew 340
 Orwant, Jon 340
 Pakin, Scott .. 348, 349
 Partl, Hubert 346
 Patashnik, Oren v, 219, 349
 Peek, Jerry 339
 Phoenix, Tom 340
 Plaice, John 332
 Podar, Sunil .. 208, 209
 Polk, Jeff 307
 Radhakrishnan, CV 216
 Rahtz, Sebastian . iv, v, 292, 342
 Raichle, Bernd 349
 Rajagopal, CV 216
 Raymond, Eric 340
 Reckdahl, Keith ... 348, 350
 Ritter, Robert 345
 Robbins, Arnold ... 340
 Rokicki, Tomas 38
 Rose, Kristoffer 217
 Rosenblatt, Bill ... 339, 340
 Rowley, Chris ... v, 347, 349
 Ryu, Young 81, 165, 328
 Séroul, Raymond .. 201
 Samarin, Alexander 342
 Schöpf, Rainer ... v, 82, 150, 347, 349
 Schlegl, Elisabeth .. 346
 Schmidt, Walter ... 350
 Schwartz, Randal L. 340
 Siebenmann, Laurent .. 201
 Smith, Brian 338
 Stallman, Richard iv, vi
 Stephenson, Peter .. 339
 Steward, Sid ... 37, 338, 341
 Strang, John 339
 Talbott, Steve 340
 Thành, Hàn Thế ... 327
 Thorup, Kresten ... 349
 Thorup, Kresten Krab . 307
 Todino, Grace 339
 Trinkle, Daniel 338
 Vesperman, Jennifer 340
 Vojta, Paul 337
 Volovich, Vlandimir 349
 Wall, Larry 340
 Wicks, Mark . v, 30, 349
 Williams, Graham . 346
 Williams, Thomas . 217, 338

Wolinski, Marcin ... 349
 Wooding, Mark 63, 349
 Zandt, Timothy ... 215,
 270, 273, 350
 Zuhn, David ... 307
 秋田純一 v
 阿部昌平 185
 阿部紀行 5
 生田誠三 342
 いけだやすし 339
 石坂智 338
 石塚圭樹 341
 市川順一 339
 伊藤敏 342
 稲垣徹 352
 井上祥史 342
 岩熊哲夫 346
 白田昭司 342
 内田昭宏 203
 内山孝憲 29, 351
 梅木秀雄 240
 浦壁厚郎 185
 江口庄英 343
 大石勝 351
 大澤英一 232
 大島利雄 v, 17, 29, 348,
 351
 大友康寛 5, 27, 352
 大野修一 217
 尾田晃 217
 奥村晴彦 v, 52, 81, 85,
 232, 284, 333, 335,
 342, 351
 小田忠雄 148
 乙部蔵己 v, 343, 351
 角藤亮 v, 215, 294, 348,
 351
 川原稔 341
 木村健一 v
 熊澤吉紀 352
 小関吉則 340
 齋藤修三郎 91, 332, 337,
 352
 坂上貴之 341
 佐藤竜一 339
 嶋田隆司 343
 竹野茂治 217, 352
 趙珍煥 v, 30, 348, 349
 土村展之 5, 334
 中川仁 27
 永田善久 352
 中野賢 v, 4, 343
 野村直 339
 野村昌孝 v, 346, 348

原信一郎 340
 平田俊作 v, 30, 348
 広瀬雄二 .. 26, 296, 337,
 351
 福井玲 337
 藤田眞作 . 217, 342-344,
 351
 古川徹生 346
 古川泰弘 342
 堀田耕作 352
 本田知亮 343
 松井正一 349
 まつもとゆきひろ 341
 皆本晃弥 341
 向井領治 342
 山賀正人 217, 351
 吉永徹美 ... v, 284, 344,
 352

す

図 259
 ——の中央揃え 171
 ——の見出し 259
 ——目次 259
 数学関数 122
 数学記号 113, 132, 165
 数字 148
 アラビア—— 149
 全角の—— 148
 ローマ—— 149
 数式
 ——中の角括弧 145
 ——中の空白の調節 119
 ——中の波括弧 145
 ——中の太字 87
 ——中の丸括弧 145
 ——の位置 85
 ——の組版 113
 ——の書体の変更 117
 ——の中の文章 118
 ——の左揃え 115
 ——の表示形式の調整 ..
 130
 ——の太字 142
 ——番号の位置 85
 ——モード 113
 番号付きの—— 115
 複数行の番号付き—— ..
 117
 文中—— 114
 別行—— 114
 数式モード 113
 数値演算 338
 スキップ 263

スコープ 100
 変数の—— 100
 スタイル
 行番号の—— 289
 索引—— 22
 参考文献—— 22
 文献一覧の—— 219
 ページ—— 250
 スタイルファイル 21, 29
 ストリームエディッタ 196
 図表
 ——見出し 171
 スプライン曲線 204
 図見出し 171
 スモールキャピタル体 79
 図目次 50, 259
 スラッシュ付きO 61
 スラント体 79

せ

制御系 217, 338
 制御点 204
 成形 8
 ——ファイル 11
 生成文法 90
 製本 229
 西暦 254
 整列 179
 積分記号 123
 セキュリティレベル 33
 セクション 61
 節 46
 絶対的な長さ 56
 絶対パス 293
 接頭語 56
 セディラ 61
 セミコロンの 58
 全角
 ——の数字 148
 全角ダーク 66
 宣言
 ——型のコマンド 103
 宣言型コマンド 59
 センチメートル 56

そ

相互参照 104
 ——できるもの 104
 ——に関わる警告 112
 ——に必要なファイル 11
 ——の簡略化 88
 ——の工夫 109
 ——の仕組み 105
 ——のハイパーリンク 292

——のラベルの表示 .. 88
 ソースコードの—— .. 291
 別の文書との—— .. 88
 相対パス .. 293
 添え字 .. 121
 上付きの—— .. 121
 下付きの—— .. 121
 ソース .. 22
 ソースファイル .. 8, 11, 22, 83
 ——の編集 .. 7
 ソフトウェア科学会 .. 337

た

dashes
 全角—— .. 66
 波—— .. 66
 倍角—— .. 66
 ターミナル .. 8, 12
 タイ .. 61
 第 .. 259
 ダイアグラム .. 217
 大域化 .. 102
 大域変数 .. 101
 大学 .. 230
 大括弧 .. 21
 大なり .. 12
 タイプセット .. 8
 ——後の生成ファイル .. 11
 タイプライタ体 .. 79
 タイプライタファミリー .. 63
 題名 .. 46, 232
 ダイアグラム .. 159
 ダガー .. 61
 高さ
 x の字の—— .. 56
 ヘッドの—— .. 239
 本文のトータルな—— .. 242
 本文領域の—— .. 239
 ルートの—— .. 143
 竹野茂治 .. 217, 352
 多言語組版 .. 332
 多言語処理 .. 352
 多段組 .. 88, 259, 347
 ダッシュ .. 66
 縦置き .. 242
 縦書き .. 34
 縦罫線 .. 178
 縦棒 .. 12
 ダブルクオート .. 58, 63
 ダブルクリック .. 8
 ダブルスペース .. 55
 ダブルダガー .. 61
 単位 .. 56
 ——のずれ .. 56

——の接頭語 .. 56
 短音符 .. 61
 段間 .. 245
 段組 .. 85
 1—— .. 85
 2—— .. 85
 短剣符 .. 61
 単語 .. 46
 ——間の空白 .. 69
 ——の引用 .. 63
 単語間空白 .. 69
 単語間スペース .. 69
 段標 .. 61
 段落 .. 45, 46
 ——間の空白 .. 69
 ——の引用 .. 63
 段落間空白 .. 69

ち

地 .. 243
 置換行列 .. 122
 縮み率 .. 263
 中央揃え .. 71
 ——を枠で囲む .. 271
 中括弧 .. 21
 注記 .. 221
 中国語 .. 32
 注釈 .. 45, 59, 230
 ——の位置 .. 59
 ——の役割 .. 59
 中途ファイル .. 11
 長音符 .. 61
 鉤形符 .. 61
 趙珍煥 .. v, 30, 348, 349
 著作権記号 .. 258
 著作権法 .. 219
 著作物 .. 219
 著者名 .. 232
 ——の頭文字 .. 223
 ——の姓名のあいだ .. 223
 ——の統一 .. 222
 ——の読み .. 232
 チルダ .. 12, 61

つ

月 .. 254
 土村展之 .. 5, 334

て

ディエレスス .. 61
 体裁 .. i
 ——の調整 .. i
 定数 .. 132, 133

定理 .. 88
 ディレクトリ .. 12
 ——の移動 .. 13
 ——の新規作成 .. 13
 ——の作り方 .. 12
 親—— .. 14
 カレント—— .. 14
 データベースキー .. 230
 テキストエディッタ .. ii, 8
 テキストファイル .. 7
 テキストモード .. 113
 デバイス .. 281
 ——ドライバ .. 24, 333
 テラ .. 57
 天 .. 243
 点 .. 136
 ——のない i .. 61
 ——のない j .. 61
 句読—— .. 57
 制御—— .. 204
 転載 .. 219
 電子情報通信学会 .. 337
 電子文書形式 .. 30
 点線 .. 209
 電卓 .. 32
 天地 .. 244
 電流 .. 57

と

統計解析 .. 338
 投稿論文 .. 337
 等幅 .. 203
 通し番号 .. 48
 ドキュメントクラス .. 82
 ——オプション .. 21, 24, 31
 読者 .. 219
 特殊記号 .. 62
 特殊文字 .. 60
 独立行政法人 .. 34
 ドラフト .. 85
 ドル .. 12

な

中川仁 .. 27
 長さ .. 57, 263
 ——の単位 .. 56
 1列の—— .. 127
 可変長の—— .. 263
 固定長の—— .. 263
 絶対的な—— .. 56
 永田善久 .. 352
 中野賢 .. v, 4, 343
 ナノ .. 57
 名前 .. 229

括弧弧 12, 21, 101, 145
 ——で囲まれた引数 .. 20
 ——の数式での出力 145
 ——の役割 20
 数式モード中の—— . 114
 文献リスト中の—— .. 223
 波ダージ 66
 並び替え 222
 文献の—— 222
 難解な用語 45
 ナンバー 12

に

二項演算子 122
 二重引用符 21
 二重かぎ括弧 20, 65
 二重括弧 65
 二重短剣符 61
 日本エディタースクール .. 345
 日本語 32
 ——の幅 56
 日本語フォント 32
 日本物理学会 337
 入力 9
 英数—— 9
 全角—— 9
 日本語—— 9
 半角—— 9
 入力支援 12
 入力通りの文字の出力 62
 任意引数 20
 認知科学会 337
 認知心理学 178

ね

熱力学温度 57
 年 254
 年月日 255

の

ノド 243
 伸び率 263
 野村直 339
 野村昌孝 v, 346, 348

は

場合分け 127, 145
 バージョンコントロール ... 306
 バージョン情報 10
 パーセント 12
 パーレン 21, 58
 倍角ダージ 66
 媒体 76

ハイパーリンク 28
 ハイフネーション 66
 ハイフン 66
 配列
 ——の要素の区切り .. 99
 バウンディングボックス ... 30,
 189, 190, 199
 波音符 61
 博士論文 229
 箱 265
 ——の上げ下げ 269
 ——の再利用 268
 ——の保存 268
 ——の用意 268
 広範囲な—— 267
 枠付きの—— ... 270, 283

パス

絶対—— 293
 相対—— 293
 パスワードによる保護 33
 破線 209
 ハチエック 61
 バックグラウンド 8, 17
 バックスラッシュ iii, 12, 20, 63
 パッケージ 21, 82
 ——オプション 25
 abstract 313
 afterpage 87
 alltt 62
 amsbsy 142, 150
 amscd 150, 159
 amsfonts . 136, 150, 165
 amsgen 150
 amsmath .. 87, 118, 128,
 142, 150, 161, 162,
 350
 amsopn 150
 amssymb . 118, 136, 150,
 165, 198
 amstext 150
 amsthm 138
 amxtra 150, 161
 array 87
 asmcld 150
 avant 328
 babel 87, 348
 balance 260
 bm 87, 142
 bookman 328
 booktabs 178
 calc .. 87, 241, 271, 307,
 308
 charter 328
 cite 234, 235

cmmt 63
 color .. 87, 215, 281, 282
 comment 61
 courier 328
 dcolumn 87, 179
 delarray 87, 129
 DocStrip 84
 doublespace 55
 eepic 186, 198, 209
 ehis 337
 emath 352
 enumerate ... 74, 75, 87
 epic . 186, 198, 208, 209
 eucal 151
 exceltex 186
 fancybox 270, 273
 fancyhdr 250, 350
 fancyheadings 250
 fontenc 60, 256, 330
 ftnright 87
 geometry . 240, 241, 245,
 247
 graphicx 87, 348
 helvet 328, 330
 hhrline 87
 his 337
 hyperref vi, 42, 44, 292,
 350
 ifthen 309
 indent 54
 indentfirst ... 54, 88, 322
 jarticle 333
 jbook 333
 jreport 333
 jsarticle 333
 jsbook 333
 jsclasses . 5, 31, 241, 333
 labelfig 201
 latexsym 136
 layout 88, 237
 leftidx 122
 Listings 291
 listings 284, 287
 lmodern 328, 330
 longtable .. 87, 181, 183
 mathpazo 81, 328
 mathptmx 81, 328
 morisawa 333
 multicol ... 88, 260, 261,
 263, 264
 multirow 180
 newcent 328
 okumacro ... 66, 75, 333
 okuverb 333

OTF 91, 332, 335
 pdftricks 216
 pict2e 186, 211
 pst-all 215
 pst-col 215
 PSTricks 198, 215
 psricks 350
 pxfonts ... 81, 142, 165,
 257, 328, 330, 331,
 337
 setspace 55
 showkeys 88, 112
 subfigure 200
 syntonly 82
 tabularx ... 87, 184, 185
 tex4ht 40
 textcomp .. 57, 256, 330
 theorem 88, 138
 tools 87
 txfonts ... 81, 142, 165,
 257, 328, 330, 331,
 337
 type1cm .. 328, 330, 332
 typelec ... 328, 330, 337
 url 44, 233, 292
 UTF 332
 varioref 88
 verbatim 88, 337
 xr 88
 xspace 88, 281
 発行 230
 発行月 230
 発行年 221, 230
 バッチ処理 4
 バッチモード 17
 ハット 12
 発表年 232
 幅
 M の字の— 56
 画像の— 190
 行列の— 127
 参考文献リストの— 220
 字下げの— 53
 箱の— 266
 表の— 87
 傍注の— 239
 本文の— 239
 本文のトータルな— 242
 文字の— 79
 要素の— 143
 パラグラフ 61
 原信一郎 340
 パラメータ 56
 —文字 99

版 7, 230, 232
 半角カナ 91
 版管理 306, 340
 番号
 —の深さ 51
 PDF しおりの見出し—
 294
 箇条書きの— 73
 脚注の— 177
 行— 285
 参考文献の— 222
 数式の— 115
 図の通し— 172
 図表の通し— 172
 図表見出しの通し— 106
 通し— 48
 表の通し— 172
 複数行の数式の— 117
 ページ— 249
 見出しの通し— 106
 論文誌の— 230
 番号付き数式 115
 版面 237, 244
 —の設定 240
 凡例 ii

ひ

日 254
 引数 13, 20
 角括弧で挟まれた— 20
 コマンドの— 20
 波括弧で挟まれた— 20
 任意の— 20
 ハイフンで始まる— 13
 必須の— 20
 ビコ 57
 非常に長い表 87
 左揃え 71
 —を枠で囲む 271
 数式の— 115
 日付 46, 254
 —の表示 255
 必須引数 20
 ビットマップ画像
 可逆圧縮— 22
 フルカラー— 22
 無圧縮— 22
 否定
 演算子の— 134
 ヒューマンインタフェース学会
 337
 表 259
 —中の小数点 87
 —の罫線 87

—の中央揃え 171
 —の見出し 259
 —目次 259
 長い— 87
 ページを跨ぐ— 87
 秒 57
 描画
 円の— 206, 211
 折れ線の— 209
 楕円の— 206
 点線の— 209
 破線の— 209
 表組み 172, 178
 欧文の— 172
 表罫線 172
 表示形式 249
 表紙の作成 323
 標準エラー出力 31
 表題 45, 229, 230
 —の役割 46
 同ページの— 85
 独立ページの— 85
 表紙の— 323
 表見出し 171
 表目次 50, 259
 平田俊作 v, 30, 348
 ビリウド 58, 180
 全角の— 57
 広瀬雄二 .. 26, 296, 337, 351
 品詞 96

ふ

部 46, 259
 —の見出し 259
 ファイル
 \$texmf 19
 —入出力 ii
 —の情報表示 12
 —の名前の変更 12
 article.cls 84
 bk10.clo 84
 bk11.clo 84
 bk12.clo 84
 book.cls 19, 84
 cid-x.map 33
 classes.dtx 84
 classes.ins 84
 config.ps 39
 DVI— 11
 dvipdfmx.def 187
 eclepsyf.sty 337
 epsbox.sty 337
 errortest.tex .. 14, 16
 fontdef.dtx 278

- gnu-head.pdf 191
jarticle.cls 19
jplain.bst ... 225, 226
jsize10.clo 26
kinsoku.tex 26
listfile.tex 26
lmodern.sty 328
ls-R 336
ltboxes.dtx 265
ltplain.dtx 277
makebb 195
Makefile 27, 298
Map—— 33
PDF—— 295
pldefs.ltx 26
proc.dtx 84
report.cls 84
Rplots.pdf 198
size10.clo 84
size11.clo 84
size12.clo 84
slides.dtx 84
test.tex 9
texmf.cnf 19
印刷できる—— 11
画像—— 172
クラス—— 237
スタイル—— 29
ソース—— 8
中途—— 50
ドキュメントクラス—— 21
フォントマップ—— 30
文献スタイル—— 230
文献データベース—— 223
文書クラス—— .. 21, 115
目次用の—— 50
ログ—— 11
ファインダー 189
ファミリー 79
 サンセリフ—— 79
 タイプライタ—— 79
 ローマン—— 79
フォルダ 12
 ——の移動 12
 ——の新規作成 12
 ——の作り方 12
フォント
 ——設定ファイル 33
 ——のアウトライン化 188
 ——の大きさ 79
 ——の選択方法 349
 ——のデザイン 213
 ——の名前 350
CID—— 32
EC—— 348
GothicBBB-Medium .. 34
Ryumin-Light 34
TrueType—— 34
Type1—— 204
Unicode—— 42
東風—— 34
さざなみ—— 34
低解像度のビットマップ
 —— 33
 日本語—— 32
 ビットマップ—— 36
 和田研—— 34
フォントの識別 ID 36
フォントマップファイル 30
付加情報 230
不許可 32
福井玲 337
複合条件判断 311
複数行数式 116
複数行番号付き数式 117
複数段落の引用 64
藤田眞作 . 217, 342-344, 351
藤田本 351
伏字 L 61
フッター 240
物理学 337
物理量 57
符点 61
浮動体 172
太字 142
 ——のページ番号 321
 数式中の—— 87
ブラックボードボード体 . 118
フランス国立コンピュータ科学・
 制御研究所 338
ブリアンブル 24, 224
 ——コマンド 19
古川徹生 346
古川泰弘 342
ブレイクポイント 17
プレビュー 4, 8, 199
 Mac OS X での—— 29
 Red Hat での—— 17
 Unix 系 OS での—— . 17,
 29
 Windows での—— 17, 29
プレビューア 17
付録 259
 ——の見出し 259
付録の追加 278
プログラミング言語 4
プログラム
 Acrobat Reader 32
 Adobe Acrobat 197-199
 Adobe Reader .. 30, 32,
 338
 Aleph 332
 \LaTeX 87, 150
 \TeX 87, 150
 APT 336
 bash 339
 bib2xml 338
 \BibTeX 219
 \BibTeX2HTML 338
 Calc 185
 \Calc2L\TeX 185
 cd 12, 13
 \ConTeXt 37
 copy 12
 cp 13
 CreateBB 23
 CVS 307
 del 12
 \DeTeX 338
 Dia 338
 dir 12
 dviout 17
 Dvipdfm 30, 245
 Dvipdfmx . 30, 216, 338
 dvips .. 38, 39, 215, 245
 Easy \TeX 27
 ebb 23, 189
 Emacs 26, 296, 298, 337
 epstopdf 194, 216
 $\varepsilon\text{-TeX}$ 327
 Excel 185, 186
 Excel2tabular 185
 file 189
 Foxit Reader 30
 Ghostscript 33, 38
 GIMP 338, 342
 Gnuplot .. 26, 217, 338,
 341, 342, 351
 Grapher 199
 GUI for PDFTK 37
 gzip 299
 help 12, 13
 ht 40
 Hyper \TeX 30
 identify 189
 Illustrator 198, 204
 ImageMagick 39
 info 13
 JabRef 236
 \jBibTeX 219, 222
 jmakeindex 44
 Keynotes 199

- Kpathsearch 19
 kpsewhich 19
 Lambda 332
 Lamed 332
 L^AT_EX 4
 L^AT_EX2HTML 338
 L^AT_EX2RTF 338
 latexmk 225, 306
 less 13, 19
 ls 13
 LyX 338
 MacOS X WorkShop . 5
 Make . 27, 225, 296, 306
 make 340
 MakeIndex 22
 man 13
 Mathematica 199
 MATLAB 198, 199, 341
 Meadow 340
 mendex 22, 44
 METAFONT... 198, 213
 METAPOST... 23, 198, 213
 Microsoft Office 3
 mkdir 12, 13
 mktexlsr 35, 336
 move 12
 mv 13
 Mxdvi 29
 Ngraph 338
 Octave .. 198, 217, 338, 341
 Omega 332
 OmniGraffle 199
 OpenOffice.org .. 3, 339
 Pages 199
 pdfcrop 194
 pdf_ε-L^AT_EX 327
 pdf_ε-T_EX 327
 pdffonts 35
 pdfimages 35
 pdfinfo 35, 36, 194, 296
 pdfL^AT_EX .. 28, 245, 327
 pdfT_EX 245, 327
 PDFtk 37, 338
 pdftops 35, 37
 pdftotext 35, 37
 pdvips 38
 Perl . 186, 196, 306, 340
 Photoshop 198
 PIC 212
 PicT_EX 198
 plain T_EX 348
 plain2 203
 PostScript 38
 PrimoPDF 197
 ps2jpdf 194, 338
 ps2pdf 194
 pst2pdf 216
 psutils 39
 pxdvi 17
 R 198, 338
 Ref for Windows ... 236
 rm 13
 Ruby 341
 SciLab 217
 sed 196
 Susie 187
 Synaptic 5, 336
 t4ht 44
 teT_EX 337
 T_EX 3
 texdoc 337
 texexec 37
 texhash 336
 Texinfo 284
 T_EXmacs 338
 T_EXShop 27
 Tgif .. 26, 198, 338, 342
 Tpic 212
 TtH 39
 Vine Linux 26, 339
 Word 3, 328
 Writer 3
 xdvi 17, 337
 xdviK 337
 X_εT_EX 327
 Xfig 198
 xml2bib 338
 Xpdf 30, 35
 X_ε-pic 217
 X_εM_εT_EX 217
 YaT_EX 26, 296, 337
 エクスプローラ 189
 ファインダー 189
 プレビュー 199
 メモ帳 8
 プログラムリスト 62
 プロット 338
 プロット 215
 分 254
 文 46
 —の引用 63, 64
 分音符 61
 文間空白 69
 文献 236
 —の管理 236
 文献スタイル 236
 abbrev 230
 alpha 230
 jabbrv 230
 jalpha 230
 jplain 230
 junrst 230
 plain 230
 unrst 230
 文献データベース 223
 文献の種類
 article 229
 book 229, 232
 booklet 229
 inbook 229
 incollection 229
 inproceedings ... 229, 233
 manual 229
 masterthesis 229
 misc 229
 phdthesis 229
 文書 i, 45
 —の概略 48
 —の執筆 i
 —の正確性 3
 —のページレイアウト .. 88
 —の末尾 220
 —の論理構造 i
 —を綴じる 244
 組版後の— 8
 体裁の整った— i
 でたらめな— 1
 ビジネス— 72
 マークアップ言語による
 — 4
 ワープロソフトによる— 3
 文章
 —表現 2
 文章表記 1
 文書クラス 83
 —オプション 24
 分数 123
 —の書き方 130
 連— 130
 文中数式 113
 〽
 米国数学会 87, 150
 ページ
 —記述言語 22, 28
 —スタイル 250
 —の拡大 30
 —の行数 239

—の区切り 253
 —の最下部 172
 —の最上部 172
 —の先頭での空き . 254
 —の背景色 283
 —の範囲 31
 —の番号 249
 —の末尾での空き . 254
 —のような箱 267
 —の余白 237
 —レイアウト 237
 —を逆順にする 31
 —の再配置 39
 改— 253
 表題— 85
 ベクトル 206
 —記号 134
 ベクトル画像 23, 204
 ベジェ曲線 204
 べた書き 62
 別行数式 113
 ヘッダー 240
 ヘルプ
 簡易の— 13
 編集 7
 原稿の— 7
 ソースファイルの— 7
 編集者 230
 変数 iii, 56, 100, 133
 変体文字 132, 169
 偏微分記号 140

ほ

ポイント 56
 法 122
 傍注 59, 242
 —と本文の空き 245
 —の幅 244
 ボールド体 79
 補足情報の追加 45
 補足説明 21
 堀田耕作 352
 本田知亮 343
 本づくり 346
 ボンド 61
 本の名前の引用 65

ま

マークアップ ... 28, 76, 78, 82
 マイクロ 57
 マイナス 66
 まえがき 52
 前書き部分 24
 マクロ 21, 83

—の再定義 92
 —の作成 140, 344
 —の定義 92
 —パッケージ 83
 数学系の— 87
 パッケージ化された— ..
 21
 マクロパッケージ 21
 松井正一 349
 まつもとゆきひろ 341
 マニュアル 229
 —配布 30
 丸括弧 21, 58, 145
 —の数式での出力 145
 丸数字 75

み

右揃え 71
 —を枠で囲む 271
 見出し 45, 48
 —の作成 48
 —の直後 88
 —の通し番号 51
 —の深さ 49
 —の変更 257
 目次用の— 49
 見出し一覧 49
 ミディアム体 79
 皆本見舛 341
 見開き 245
 苗字 229
 ミリ 57
 ミリメートル 56
 明朝体 34, 80

む

向井領治 342
 無効文字 99

め

名詞 96
 明度 281
 命令 20, 92
 —型のコマンド 103
 装置に依存した— 28
 デバイス依存の— 28
 メートル 57
 メガ 57
 メモ帳 8
 面付け 39

も

モード 113

数式— 113
 テキスト— 113
 目次 45, 49, 50, 259
 —の作成 105
 —の番号付けの深さ 51
 —の深さ 51
 —の見出し 259
 —の見出しの変更 . 257
 —用の中途ファイル 22
 —用の見出し 49
 secnumdepth 51
 tocdepth 51
 図— 50, 259
 表— 50, 259
 文字 3, 46, 76
 —間空白 68
 —サイズ 85
 キリル— 349
 半角カナ 91
 文字サイズ 86
 無視される文字 99
 文字集合 332
 モリサワ 34
 モル 57

や

矢印 135, 167, 206
 野鳥 337
 山賀正人 217, 351
 山括弧 21

ゆ

ユニコード 332, 352
 —エンコーディング . 36

よ

揚音符 61
 用語 45
 —の統一 280
 用紙 237, 241
 —の既定のサイズ . 241
 —の空白 237
 —の高さ 241
 —の幅 241
 —の左端 243
 —の方向 31
 —の右端 243
 —の大きさ 85
 —の大きさの指定 . 30,
 38
 —のサイズ 85
 —の方向 85
 用紙サイズ 86

ヨーロッパ語圏 328
 抑揚音符 61
 横置き 31, 242
 横書き 34
 横罫線 128, 175
 吉永徹美 ... v, 284, 344, 352
 余白 237
 上下左右の—— 240
 読み 232
 予約文字 60, 91

ら

ラベル 104

り

リーダー 277
 リダイレクト 31
 両面印刷 240, 244
 リング 61

れ

列指定子 127, 174
 行列における—— 127
 小数点を揃える—— .. 180
 表中の—— 174
 レポート
 科学技術系の—— i
 レポート作成 86
 連分数 130, 164

ろ

ローマ数字 149, 251
 ローマン体 79
 ログファイル 11, 22
 ロシア語 349
 ロング 61
 ——ハンガリアン 61
 論文
 ——原稿 337
 ——作成 220
 ——投稿 33

——における図表 ... 175
 会議録中の—— 229
 科学技術系の—— i
 修士—— 229
 投稿—— 337
 博士—— 229
 論文誌 229, 230, 232
 ——の巻 232
 ——の番号 232

わ

ワープロソフト 3
 枠
 ——と文字の間隔 ... 266
 ——の色 284
 ——の太さ 266
 2重の—— 270
 和田研フォント 34
 和文
 ——の引用符 21
 和暦 254

好き好き L^AT_EX 2_ε 初級編

© 渡辺徹 2004, 2005, 2006, 2016

発行日	2004/04/02	ver. 0.10	配布
	2004/04/16	ver. 0.20	配布
	2004/04/30	ver. 0.21	配布
	2004/08/05	ver. 0.30	配布
	2004/10/14	ver. 0.31	配布
	2004/12/28	ver. 0.32	配布
	2005/02/15	ver. 0.33	配布
	2005/03/20	ver. 0.34	配布
	2006/04/20	ver. 1.00	配布
	2006/05/08	ver. 1.10	配布
	2006/05/12	ver. 1.11	配布
	2006/05/12	ver. 1.12	配布
	2006/08/18	ver. 1.13	配布
	2016/07/23	ver. 1.14	配布
