

L^AT_EXによる論議 の手引き

2006年度版 改訂 3版

```
\begin{titlepage}%
\parindent=\z@
\let\footnotesize\small
\let\footnoterule\relax
\let\footnote\thanks
\par\vskip2\Cvs
{\Huge\gtfamily\sffamily\@title}%
\par\vskip\fbboxsep
\hrule\height1ex
\par\vskip\tw@\fbboxsep
{\Large\@date}%
\par\vskip\Cvs\vfill
\hb@xt@\fullwidth{%
\includegraphics*{images/mytitle}\hss}%
\ifx\@contact\@empty\@thanks\vfill\null
\else\vfill\begin{large}\@contact\end{large}%
\par\vskip2\Cvs\fi
\end{titlepage}%
```

公立はこだて未来

システム情報部

FUNNIST 編集員

監査

Copyright © 2003, 2004, 2006 FUNNIST
Copyright © 2003, 2004, 2005, 2006 渡邉

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

<http://www.gnu.org/copyleft/fdl.ja.html>
で参照される。

本冊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所有者の商標または商標登録であり所有物です。本文中では ™ 及び ® は明記していません。

まえがき

これは何のための冊子か

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

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

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

LaTeXの導入やその周辺の情報に関しては、<http://oku.edu.mie-u.ac.jp/~okumura/texwiki/>

等を参照してください。

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ことをお勧めします。

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ことをお勧めします。

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

FUNNIST について

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

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

凡例

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

書体	意味	例
ロマン 体	通常の文章	dvi pd fm
サンセリフ体	パッケージ (3.21 節)	dvi pd fm
タイポ 体	キーボードの入力など	dvi pd fm
イタリック体	変数や強調	dvi pd fm
スラント体	オプション (3.21.2 節)	dvi pd fm

本文中で左側にタイプライタ体、欄 にそれに準じた出力例があるものは、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...
--	--

テキストエディタなどを使い、欄 ファイルで左側のように入力すると、欄 の出力例と同じような結果を確認できます。

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

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

先頭のドル‘\$’はコン欄 表示されている記号で、欄 入力しません。

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

何らかの文欄 や数値に置き換わるものは (変数) のように表記しています。

謝辞

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

T_EX の作者である Donald Knuth 氏，L^AT_EX の作者である Leslie Lamport 氏，L^AT_EX 2_ε の開発をされた Frank Mittelbach 氏，Johannes Braams 氏，David Carlisle 氏，Michael Downes 氏，Alan Jeffery 氏，Sebastian Rahtz 氏，Chris Rowley 氏，Rainer Schöpf 氏，T_EX 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 pT_EX を移植して下さった角藤^{あきら}亮氏，Dviout を開発された大^{あきら} と乙部^{よしき}巖己氏，BrT_EX 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，dvi_{pdfm} の作者である Mark Wicks 氏，Dvipdfm_x の保守管理 をされておられる平^{あきら} 作氏 と Cho Jin-Hwan 氏，PostScript や PDF などの^{あきら} 記述語 を作成された Adobe 社の方々，フリーウェア・マクロパッケージなどの作成で，T_EX の分野において貢献された方々にも感謝いたします。

さらに，この冊子を直接ご覧いただき，ご助言を戴いた大友康寛氏，串健太氏，^{あきら} 善氏野村^{あきら} 三直^{あきら} 茂和^{あきら} 忠^{あきら} 儀氏 には感謝しております。^{あきら} 階 での不備や不^{あきら} を指摘していただき，この冊子の作成が円滑に進み，また主^{あきら} になりすぎていた部分も見直すことが出来ました。

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

変履歴

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

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

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

2003 年 11 月

『L^AT_EX による論 彙 の手引き』という名前で、主に大 擧 で冊子を配布する。

2004 年 4 月 2 日

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

2004 年 4 月 16 日

2004 年 4 月 2 日の誤 謬 を修正し version 0.2 を配布する。

2004 年 4 月 30 日

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

2004 年 8 月 5 日

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

2004 年 8 月 19 日

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

2004 年 9 月

上記入門書の簡易版であり、論 彙 作成 に特化し、題 名 を改め『L^AT_EX による論文作成の手引き』を配布する。これは上 記 書 の派 物 であり、入 書 とは別物という位置 けで発行した。論 彙 に直 接 に必要ないと思われる部分を削除した。

2005 年 3 月 20 日

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

2006 年 3 月

最近の動向を改訂 3 版に反映しました。

目次

まえがき	i
謝辞	iii
第 1 章 論 檪 を始める前に	1
1.1 論文とは何か	1
1.2 組版とはなんだろうか	1
1.3 文 檪	1
1.4 TeX とは何か	3
1.5 WYSIWYG とは何か	3
1.6 一纏 とは何か	3
1.7 L ^A T _E X とは何か	3
1.7.1 L ^A T _E X の導入	4
1.7.2 情報の入 檪	4
第 2 章 L ^A T _E X の基本	5
2.1 基本の基本	5
2.1.1 処理の流れ	5
2.1.2 動かしてみる	6
2.1.3 原稿 檪 の注意	8
2.1.4 フォルダ・ファイルの基本的な操作	8
2.1.5 エラに遭遇する	9
2.1.6 プレビューアの操作	12
2.1.7 コマンド	14
2.1.8 括弧について	14
2.2 L ^A T _E X に関わる 檪 形式	15
2.3 コマンドの基本	16
2.3.1 プリア 檪	16
2.4 執筆における基本	18

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

3.21.1	標準的なクラス	43
3.21.2	クラスオプション	44
3.21.3	標準で使用できるパッケージ	45
第4章 参考文献の出力		47
4.1	参考文献の明記	47
4.2	参考文献を手動で並べる場合	48
4.2.1	文献の並べ方	49
4.3	参考文献をプログラムで並べ替えるとき	50
4.3.1	JBIBTEX の使い方	50
4.3.2	文献データの作成	50
4.3.3	参考文献の出力	51
4.3.4	文献の種類と項目	52
4.3.5	各文献の出力例	53
4.3.6	文献の追記	54
4.3.7	文献を同時に複数しているとき	57
4.3.8	参照の形式を変更する	57
第5章 原稿の出力		59
5.1	出力の種類	59
5.1.1	DVI	60
5.1.2	PostScript	60
5.1.3	PDF	60
5.2	DVIをPDFに——Dvipdfmx	61
5.3	DVIをPSに——dvipsk	62
第6章 コマンドとマクロ		63
6.1	マクロ言語とは?	63
6.2	記号とコマンド	63
6.2.1	記号の分類	63
6.2.2	コマンド	64
6.2.3	コマンドの定義	65
6.2.4	文字やコマンドの区切り	66
6.2.5	コマンドの引数	68
6.3	グルーピングと入れ子構造	69
6.4	宣言と命令の違い	69
6.5	相違	71
6.5.1	相違の仕組み	71

6.5.2	カウタ	72
6.6	相懸の工夫	73
6.6.1	参照ラベルの表示 showkeys	76
6.6.2	相懸に関わる L ^A T _E X の警告	76
第7章 数式の書き方		77
7.1	はじめに	77
7.2	数式の出力	77
7.2.1	文獻	77
7.2.2	グルピング	78
7.2.3	別裁	78
7.2.4	番付 き数式	79
7.2.5	複裁	79
7.2.6	複裁 き数式	79
7.3	書体の変更	80
7.4	数式における空白の調節	81
7.5	基幹 な数式コマンド	81
7.5.1	添え字	82
7.5.2	数懸	83
7.5.3	大きさ可変の数懸	83
7.5.4	区切り記号と括弧	84
7.5.5	行列	86
7.6	表懸の調整	89
7.7	数式モード中の記号	90
7.7.1	ギリシャ文字	90
7.7.2	関係 や演算子などの数懸	92
7.7.3	標準ではない数懸	94
7.8	定義や定理など	94
7.8.1	定環境のカスタム	95
7.9	雑多なこと	96
7.9.1	記号の積み重ね	98
7.9.2	記号の重ね合わせ	98
7.9.3	数式の太字	99
7.9.4	高さを揃える	99
7.9.5	スマク 分数の書き方	100
7.9.6	場合 け	101
7.9.7	数式モード中の空白と書体	101
7.9.8	ダイヤグラの 例	102

第 8 章	図表の構成	103
8.1	図表の基礎	103
8.1.1	一般的な取り決め	103
8.1.2	L ^A T _E X での扱い	104
8.2	表	105
8.2.1	表中の脚注	107
8.3	書籍スタイルの表線 booktabs	107
8.4	小断え——dcolumn	109
8.5	表における行の連結 multirow	109
8.5.1	表断え ツル	110
8.6	図に関する制約と画像の扱い	111
8.7	画像ファイルの張り込み	111
8.7.1	デバネ断えラ 選択	112
8.7.2	具体的な手順	113
8.7.3	張り込みにおけるオプション	114
8.7.4	画像の拡大や回転等の操作	116
8.7.5	Dvipdfmx における EPS 画像の扱い	117
8.7.6	dvips と Dvipdfmx の併用	118
8.7.7	レポート・論文 における図の張り込み	118
8.7.8	汎断えな画像の作成と活用	119
8.7.9	プログラム特有の処理	119
8.8	図の張り込みの際の工夫	121
8.8.1	図を二つ横に並べる	121
8.8.2	画像に文字を追加する——labelfig	122
8.9	その他の描画に関する情報	123
8.9.1	化学断え学	123
8.9.2	グラフの描画	123
8.9.3	Xy-pic	124
第 9 章	L^AT_EX の応用	125
9.1	ページ断えウ 簡単な設定	125
9.1.1	版面の断えウ	125
9.2	レイア断え 制御	127
9.3	あらかじめ定義されている見出しの変更	128
9.4	多段組	128
9.5	箱の操作	129
9.5.1	枠のない箱	129
9.5.2	枠のある箱	130

9.5.3	広範な箱	130
9.5.4	罫線と下線	131
9.6	空白の挿入	132
9.6.1	水平向の空き	132
9.6.2	垂直向の空き	133
9.7	付録の追加	134
9.8	原稿を複数のファイルに分ける	134
9.9	翻	135
9.10	用語の統一	135
9.11	URL の記述	136
付録 A 最近の動向		137
A.1	PDF と TeX	137
A.2	文字と書体	137
A.3	日翻 クラスファイル	138
A.4	画像やグラフィクス 周辺	138
A.5	今後について	138
付録 B 論文のサンプル		139
B.1	中巻のサンプル	139
B.2	学誌のサンプル	144
付録 C GNU Free Documentation License		151
1.	APPLICABILITY AND DEFINITIONS	151
2.	VERBATIM COPYING	153
3.	COPYING IN QUANTITY	153
4.	MODIFICATIONS	153
5.	COMBINING DOCUMENTS	155
6.	COLLECTIONS OF DOCUMENTS	155
7.	AGGREGATION WITH INDEPENDENT WORKS	156
8.	TRANSLATION	156
9.	TERMINATION	156
10.	FUTURE REVISIONS OF THIS LICENSE	156
	ADDENDUM: How to use this License for your documents	157
参		159
索引		161

図表

2.1	処理の流れ	5
2.2	xdvi の起働	13
3.1	テキスト入力の出力例	37
8.1	1 段組で横に図を二つ並べる	121
8.2	labelfig の使い方	123
9.1	版面の A^{A} 使用できる長さ	126
B.1	版面の A^{A} 使用できる長さ	148

表

2.1	Windows OS の基本コマンド	8
2.2	Unix 系 OS の基本コマンド	8
3.1	文書の構築	20
3.2	L ^A T _E X での見出しの定義	21
3.3	見出しの階層	22
3.4	L ^A T _E X で使用できる主な単位	25
3.5	SI の基 A	26
3.6	10^n の修飾子	26
3.7	特 A	28
3.8	アクセント記号	28
3.9	ダッシュなど	32
3.10	揃えの命令と宣言	38
3.11	文字の大きさの変更	40
3.12	基準の文字の大きさによる A 挙動の違い	40
3.13	書体を変更するコマンド	41
3.14	和 A の A	42
4.1	文献の形式	53
4.2	フィールド 名	54
4.3	文献の種類における A	55
4.4	cite パッケージ 変更できる命令	58
6.1	カテゴリ A 一覧	68

6.2	あらかじめ定義されているカタ 名	72
6.3	要素に応じたラメル 貼り方	73
7.1	数式モジュールにおける 書体の変更	80
7.2	amssymb による数罫 の拡張	81
7.3	数式における空白の制御	81
7.4	添え字の使い方の例	82
7.5	主な数罫	83
7.6	大きさ可変の数罫	84
7.7	主な区切り記号	85
7.8	括弧の大きさを指定する例	86
7.9	array 環境の主な列罫	87
7.10	array 環境中での罫線の命令	88
7.11	数式の表罫 の変更	90
7.12	ギリシャ小字	90
7.13	ギリシャ小字 の変換	91
7.14	ギリシャ大字	91
7.15	関係	92
7.16	2 項罫子	92
7.17	大罫子	92
7.18	小さいアクセント	93
7.19	大きいアクセント	93
7.20	矢印	93
7.21	特殊な数罫	94
7.22	点	94
7.23	標準ではない数罫	94
8.1	浮動 の種類	104
8.2	浮動 の位置	104
8.3	tabular 環境の主な列罫	105
8.4	tabular 環境中での罫線の命令	106
8.5	表の出力例	106
8.6	各種デバネラ 画罫	112
9.1	定義 みの見出しの変更	128
9.2	改行を許す水平向 の空き	132
9.3	改行を許さない水平向 の空き	132
9.4	垂直 の空き	133

第 1 章

論議 を始める前に

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

1.1 論文とは何か

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

1.2 組版とはなんだろうか

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

現代ではコンピュータ上で文書を組版できるようになりました。だれでも手軽に印刷用の
美しいフォントを用いた組版が可能で、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**藪** などは一定のルールに沿って整形されているもので
す。たとえば 1 行を何文字にするか 1 ページを何行にするかなどの約束事があります。この
ような様式をどのようにするのは各出版社や各種学会が一定の条件を判断して決めてい
ます。

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

1.3 文藝

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

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

次の例文の中には多くの文書の上での約束が秘められています。

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

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、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、em-dash の場合は前後に空白を入れません。

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

クオートで一文を引用していますが、`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 en-dash を用います。`楙語`でも波ダッシュ‘~’は使いません。「図3を参照せよ」という意味の‘(see fig. 3)’ですが、丸括弧（パーレン）の左側（起こし）に空白を入れていますが、`欄`（け）には入れていません。‘fig.’と‘3’の間で改行することは好ましくないため、チルダ‘~’を補っています。`戴`の等号‘=’は関係演算子を意味していますので、`籐`に適切な空白が挿入されることになります。

<p>\$\$ agenda \leftarrow office \$\$ \$\$ \mathit{agenda} \leftarrow \mathit{office} \$\$</p>	<p><i>agenda ← office</i> <i>agenda ← office</i></p>
--	--

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。`籐者`の意図としては「リスト *agenda* に *office* を代入する」ということになりますが、一つ目は `\mathit` というコマンドを使っていないために、「変数 a, g, e, n, d, a の積に変数 o, f, f, i, c, e の積を代入する」という全く異なった意味になってしまいます。

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

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

この冊子でもそのような「記号の使い方」に関する部分を取り扱い、それらを 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`束`で**強調**して表記しています。

近年はワープロソフトと呼ばれるソフトウェアが多数存在します。OpenOffice.org とか Microsoft Office などがある。これらのワープロソフトと LaTeX とには決定的な差があります。ワープロソフトは文書の要素に直接視覚的な調整を施します。例えば、「T」という文字をワープロソフトで斜体にする、罫線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。LaTeX をうまく使いこなせば、このような問題は発生しません。なぜならば、罫線に意図を付けることができるからです。

1.4 TeX とは何か

TeX [12] とは Donald Knuth 氏によって開発された組版プログラムで、筆すべきことは数式の処理に優れていること、簡単なボットの作成から論文の作成果ては商議にも耐えうる機能を持っていることなどです。

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

1.5 WYSIWYG とは何か

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

TeX 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。罫紙に印刷するのは大変時間を必要とし、なおかつ地球環境の悪化を促進するものです。そのためコンピュータの画面上で確認作業をします。これをプレビューと言います。

1.6 一括 とは何か

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

1.7 LaTeX とは何か

組版プログラムとしての TeX は完成度が非常に高く、機能です。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ことができます。このシステムを開発されたのが Leslie Lamport 氏で、彼の作成したシステムを

L^AT_EX [13] と言います。

L^AT_EX も HTML と同様のマークアップ方式を採用しています。簡便な例を挙げると、

```
<CENTER>
  人類の原理であ
</CENTER>
```

という記述があるとして、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を行います。HTML での表記が L^AT_EX では、

```
\begin{center}
  人類の原理である
\end{center}
```

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

T_EX も L^AT_EX も欧文言語圏のためのプログラムですから標準では日本語を処理することが出来ませんが、野賢氏を始めとするアスキーの方々が T_EX の日本語化をしてくださったので、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裔びーてっくされた T_EX や L^AT_EX をそれぞれ pT_EX、pL^AT_EX と呼びます。

▼ 1.7.1 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なインストールをする事になります。

Windows 阿部紀行氏 (<http://www.ms.u-tokyo.ac.jp/~abenori/>) 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。

Mac OS X MacOS X WorkShop や EasyPackage (<http://www.ie.u-ryukyu.ac.jp/darwin2/>) 等で簡単に周辺ツールも導入できます。後の展開については MacWiki (<http://macwiki.sourceforge.jp/>) 等を参照してください。

Vine Linux コンソールから root ユーザで `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。

▼ 1.7.2 情報の入巻

Leslie Lamport 氏の『文書処理システム L^AT_EX 2_ε』 [13] やコパニオンシリーズ [2-4], 入門用として奥村晴彦氏の『L^AT_EX 2_ε 美文書作成入門』 [21], それに藤田眞作氏の書籍 [9] や、鄙巖己氏と江口庄英氏による *Another Manual* シリーズ [1, 22, 23] が参考になると思います。嵐の書籍は入手が容易だと思います。

第 2 章

L^AT_EX の基本

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

2.1 基本の基本

▼ 2.1.1 処理の流れ

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

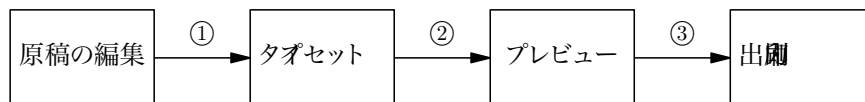


図 2.1 処理の流れ

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

1. 原稿 (ソースファイル) の編集

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

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

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

2. タオセット (DVI 出力)

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

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

場合は latex, 構語 を扱うときは latex を日本語化した platex です。シェルやマ
ンドプロンプト~~などの~~ ,

```
$ platex file.tex
```

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

3. プレビュー (羅)

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

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

などとすると良いでしょう。後 のアンド '&' があるとプログラムがバクグラウンドで 起
動しますので、欄 です。Dviout をインストールした Windows ならばダブルクリッ
クするだけで見られるでしょう。これが図 2.1 の③の矢印に対応します。

このような流れがあることを確認して実際に動くかどうかを試してみましょう。

▼ 2.1.2 動かしてみる

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

とりあえず自分のいつも使っている~~系~~ (メモ帳や Emacs など) で以下のよ
うな~~アイ~~ first.tex

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

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

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

となっていると思います。番右側 の [~] という部分が半角入力~~入力~~ (かな) か全角入力
(日本語入力) かの違いを表します。うゑの状態は英数文字の入力ができます。ここで半
角~~角~~ の入力を切り替えるためには CTRL を押しながらか (¥) 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。~~素~~ 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*2。 [あ] の状態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

*2 [Shift]+[Space] で切り替える場合もあります。

それで半角と全角の切り替えを行います。Unix系OSは違いますので注意してください。
詳しくはご自分のテキストエディタ 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 `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。そしてシェル上やコマンドプロンプトでファイルが存在するディレクトリ (Windows の方はフォルダ) に移動して

```
$ platex file.tex
```

として実行してください。すると端末 (ターミナル) に

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

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

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

```
$ ls file.*
```

とすると

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

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

`(file).aux` 次回のタイプセットに必要な **中途ファイル**。`状態` 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 した L^AT_EX の原稿である **ソースファイル**。

▼ 2.1.3 原稿端の注意

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

\$ % & _ { } ~ ^ \

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

| < > はそれぞれ — i i

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

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

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

▼ 2.1.4 フォルダ・ファイル 基本的な操作

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

表 2.1 Windows OS の基本コマンド

コマンド名	意味
mkdir	新規にフォルダ作成
cd	フォルダの移動
dir	ファイル 情報を表示
move	ファイル 名を変換
copy	ファイル 複製
del	ファイルを 削除
help	コマンドのヘルプ 表示

表 2.2 Unix 系 OS の基本コマンド

コマンド名	意味
mkdir	ディレクトリ の 作成
cd	ディレクトリ 移動
ls	ファイル 情報
mv	ファイル 名の変換
cp	ファイル 複製
rm	ファイルを 削除
help	内部コマンドのヘルプ 表示
man	コマンドのヘルプ 表示

それぞれのコマンドの使い方 (ヘルプ) は `command /?` とすることで表示できます。または [スタート] メニューの [ヘルプ] からコマンドプロンプトについて調べてみても同様のことができます。どちらかという 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すので、日ごろから使うように習慣付けをすると良いでしょう。シェルと言っても何種類かありますし、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 `command --help | less` のようにすると `less` がページを整形します。もう少し詳しいヘルプが見たいときは `man command` とします。もっと詳しいヘルプが見たいときは `info command` のようにすると `info` がページを整形します。`less` や `info` の操作方法は若干癖がありますので慣れるまで時間がかかるかもしれません。Unix 系 OS ならば

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

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

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

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

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

上記の操作はどのような結果をもたらしたと考えられるでしょうか。

新規にディレクトリ `anydir` を作成し、`現` 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` の操作でうえの階層のディレクトリ (親ディレクトリ) に `file.txt` がないことでそれを確認できます。

▼ 2.1.5 エラに 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。廻 を中断するとユーザにどうすれば良いかを促します。このときエラー 疑問 ‘?’ が表示されます。

▷ **例題 2.2** まずは以下のマクロ `error.tex` を作成してください。

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

次に `error.tex` を `platex error` でコンパイルしてください。するとエラー

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

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

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

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

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

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

修正 のマクロ 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 行目のアンド ‘&’ とドル ‘\$’ と 4 行目のサバー ‘#’ にバックスラッシュ ‘\’ を付けます。これを再びコンパイルしてみてください。戻 は

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


?
 「
 ドル '\$' の書き忘れがあるとされています。4行目のエラーメッセージで丁度アンダーバー
 ' _ ' の部分で表示が改行されていますから、この部分に間違いがあることが分かります。どう
 やらアンダーバーは '\$' 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 **リセット** 終了してください。一つ目のエラーとして

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

が表示されます。 **リセット** 中断しませんが

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

という煩雑な表示が出現します。これは **Overfull \hbox** という警告であることが分かりま
 す。次に成程 の DVI ファイル `error.dvi` を **ビュー** してください。すると行が **ページ**
 はみ出しています。程 の **アバ** 関するエラーにおいて

```
「
<inserted text> $
」
```

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

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

これで望み通りうまく行きそうです。実際に上記の **リセット** し、その結果を吟味
 してください。

LaTeX 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。この段階でこちらも疑問符 '?' で返事を返すと

```
「
Type <return> to proceed, S to scroll future error messages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
?
」
```

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

- [Enter] エラに対して L^AT_EX が適当な対処をした後にタイプセットを続行します。
- [S] [Enter] キーを押し続けたことと同じ動作をします。
- [R] エラが検出されても停止せずにノンストップで続行します。
- [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] 〈文書〉 文字列を挿入してタイプセットを続けます。元の原稿に修正は加えられません。
- [H] そのエラに対する英語のヘルプを表示します。
- [X] タイプセットを終了します。

[X] キーは余り押しはいけません。バッファ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らなくなります。

```
「
*
(Please type a command or say ‘\end’)
」
```

この場合コンソール上から

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

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

▼ 2.1.6 プレビューの操作

プレビューを行うプログラムのことを**プレビューア**と言います。OS によって使用可能なプレビューアが異なります。Windows ならば **大鶴** 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。まずシェル上で hoge.dvi の存在するディレクトリに移動し、xdvi に対して **ファイル名** を指定し、

```
$ xdvi hoge&
```

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

xdvi の基本的な操作方法を説明します。**欄** 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**が表示します。

xdvi での**マウス**はク **拡大**の機能を持っています。それぞれ

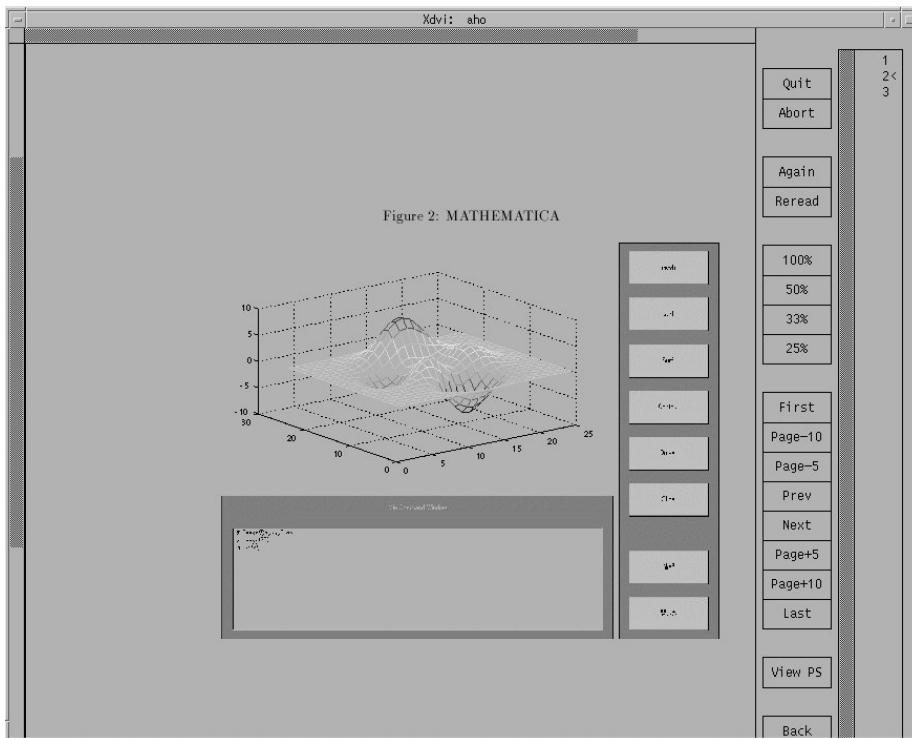


図 2.2 xdvi の起動

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、欄にある‘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 *file*.dvi を再欄 する。
 First 先頭ページ 移動する。
 Prev 前ページ 移動する。
 Next 次ページ 移動する。
 Last 最終ページ 移動する。
 View PS PostScript ファイル 見る。
 File DVI ファイル 別に開く。

終了するには‘Quit’ボタンを押します。

▼ 2.1.7 コマンド

L^AT_EX では原稿を三つの**区**に分割することができます。それに伴い、いくつかのコマンドは、特定の**区**しか**区**で使用できません。

原稿 区 分 (イニシャルコマンド) <code>\documentclass[<オプション,...>]{<クラス>}[<リリネ >]</code> 前書き部分 (プリアブルコマンド) <code>\begin{document}</code> 本文 (ボディ) <code>\end{document}</code>
--

この中で `\documentclass` 命令と `document` 環境は必須であり、絶対に必要な記述です。原稿**区**分 (イニシャル) 部分には**イニシャルコマンド** と呼ばれるコマンドを記述することができます。同じように前書き (プリアブル) 部分には**プリアブルコマンド** や定義などを記述することができます。そして、`document` 環境によって挟まれた本文部分にはコマンドの定義や組版用のコマンドを記述します。それぞれのコマンドは定められた場所で使うように決められています。ユーザがプリアブルコマンドを本文で使うことができないように L^AT_EX の内部で細工が施されています。

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

コマンド バックスラッシュ (Windows の方は円**フ**) と共に用いられる文**節**

命令 単独で使用するコマンド。 **数** を取ることができる。

例：`\alpha`, `\maketitle`

環境 ‘`\begin{<何々>}`’ と ‘`\end{<何々>}`’ で囲まれている領域 またはそれを囲むためのコマンド。 **数** を取ることができる。

例：`\begin{center}文節` `\end{center}`

引数 コマンドに受け渡す文**節**

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

例：`\section{引数}`

任意数 オプションとも言う。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[任意数]{jbook}`

▼ 2.1.8 括弧について

さて、L^AT_EX の基本を知ったところで**括弧**についての取決めをしたいと思います。**括弧** については色々な呼び方があるようですが、**誤** 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—「」 引用や会議 などに使う。

二重かぎ括弧—『』 **書籍** 中の引用などに使う。

引籍 — ‘ ’ シングルクオートとも言う。左にあるほうを左シングルクオート、右にあるほうを右シングルクオート と言う。用 に使う。

二重 — “ ” ダブルクオートとも言う。左にあるほうを左ダブルクオート、右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** **パレ**とも 言う。 **籍** の補~~カ~~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、**裏**を一つの~~カ~~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、~~カ~~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**、~~カ~~のものを**かぎ括弧**と言うことにします。 ~~カ~~ 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~~カ~~イ 形式

タオセット 時に作成される中途フォル 以外にも L^AT_EX では多くのファイル形式が存在することを経験するでしょう。般 に~~カ~~形式は拡張 によって種類を識別します。

〈~~カ~~名〉. 拡張

のように~~カ~~後文字で区別されます。

パッケージ~~カ~~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。 **裏** 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~~カ~~** 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~~カ~~** とも言う。
- .clo クラスの~~カ~~ン 応じた設定を記述した~~カ~~。
- .fd 書体の属性を定義した~~カ~~。 **ユ~~カ~~** 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 参考文献の表紙を決めるもの, 参考文献スタイル と言う.
- .eps Adobe 社が開発したページ記述言語 PostScript で書かれたファイル, 主にベクトル画像などに使われる.

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

- .log L^AT_EX の組織の詳細 ログファイル と言う.
- .aux 相変換などの情報が書かれたファイル, 1 度臨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 「表紙」を出力するための目録が書き出されたファイル.
- .lof 「図」を出力するための図録が書き出されたファイル.
- .lot 「表」を出力するための表録が書き出されたファイル.
- .bb1 Bib_TE_X によって並べ替えをした後の参考文献リスト, thebibliography 環境を用いて記述されている.
- .blg Bib_TE_X の実行結果が出力されるログファイル.

その他ファイルに関わる拡張子として主に以下があります.

- .jpg 写真などのフルカラーに適したビットマップ画像
- .bmp Windows 標準の無圧縮ビットマップ画像
- .png 可圧縮で Dvips 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 プリアンブルのコマンド

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

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

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

〈クラス名〉には 3.21.1 節で紹介するものが使えます。〈オプション〉にはそれぞれのクラスが用意している任意引数を渡すことが出来ます。このオプションのことを特に**文書クラスオプション**とか**ドキュメントクラスオプション**と言います。〈リリース〉には自分の使っているクラスがいつ 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のようになります。

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

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

他にも 3.21.3 節で紹介しているようなパッケージを使う場合は `プリアブル` 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` はそのパッケージのバージョンを指定できます。例えば、 `ファイル` などを \LaTeX で扱いたいと思い、 `デバイスドライバ` として `Dvipdfmx` を使う場合は

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

のように `graphicx` パッケージを使うことを **プリアブル** 宣言します。

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

文書クラスオプションやパッケージオプションのいずれにしても、たいいてい「命令」と「必須数」のあいだの〈オプション〉(`数` は **複数**) **することができます**。例えば

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

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

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

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

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

基的 な **変換** 次のようになります。

```
\documentclass[10pt,a4paper,oneside]{jarticle}
```

```

\usepackage[dvipdfm]{graphicx}
\usepackage[dvipdfm,usenames]{color}
\begin{document}
ここに本文を書きます.
\end{document}

```

後述の `\documentclass` 指定に関しては、`\usepackage` 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 `\usepackage` を追加するのが安全です。

```

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

```

2.4 執筆における基本

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針であるため、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 とその周辺のパッケージを活用しようと思えば、Unix 系 OS を使うと快適な執筆環境を得る事が出来ます。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。

Emacs L^AT_EX の原稿となる `tex` 編集する時に使える `tex`。

YaTeX 上記 Emacs 上で動作する広瀬氏による L^AT_EX 執筆支援システムで、完成度が高いものです。

Tgif Unix 系 OS で広く使われているベクタ画像プログラムです。

Gnuplot Unix 系 OS で広く使われているグラフを描画したり、データのためのプログラムです。

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、再コンパイルが楽になります。

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執筆環境が数多く存在します。

WinShell (Windows) Ingo H. de Boer 氏らによる統執筆環境です。コマンドラインからの煩雑な操作なしに `reset` 等が出来るようになります。

EasyT_EX (Windows) 中川氏による統執筆環境です。

T_EXShop (Mac OS X) Mac OS X で使用できる Richard Koch 氏らによる統執筆支援環境です。PDF でのプレビューが可能で `ps` における表示がきれいです。

第 3 章

文章の書き方

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

3.1 文章の論理構

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

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

目次 (contents) ページが多い場合には目次をつけて読者に参照しやすいようにします。大規模な文書の場合、論理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、論理などでは目次は必須です。

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、論理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筋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。場合によっては直後の字下げは慣習に行いません。

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。注釈はあくまで補綴であって、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。漢字の最小構成単位は**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。英語の場合は表意文字なので最小単位は**文字** (letter) に相当します。

表 3.1 文書の構成

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、罫線の配置、図の作成など、様々なことを半自動的に行ってくれます。ここではその基本的な約束を紹介します。

3.2 表題

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

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

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

```
\maketitle
```

とします。

例を示すと入力が

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

であったならば、大体の出力は以下ようになります。

はじめての LaTeX

未来太郎

2004年3月30日

LaTeX を使うのは...

3.3 見出し

文書に見出し (sectioning) と目次 (contents) がなければ、`Ctrl+F` 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、`LaTeX` 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) を付け足すのも効率的です。

▼ 3.3.1 見出しの出力

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

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

表 3.2 `LaTeX` での見出しの定義

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

*`article` や `jarticle` では定義されていません。

出しを作成します。`LaTeX` の空白の調節や改ページ、`LaTeX`、`LaTeX` の変更などはほぼ自動的に行われ、通し番号 (serial number) が付加されます。‘[(目次用の見出し)]’ という任意引数がありますが、これは見出しが非常に長いときに、それを短縮した文字列を目次に書き出すようにします。別に長いときだけでなく、見出しと目次の文節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。

```

\chapter{特論}           }           第 1 章 特論
  \section{歴}           }           1.1 歴
\chapter{一論}           }           第 2 章 一論
  \section{電気との関連} }           2.1 電気との関連
    \subsection{電気の次} }           2.1.1 電気の次

```

▼ 3.3.2 見出しの深さ

`LaTeX` ではあらかじめ部 (part)、章 (chapter)、節 (section)、小節 (subsection)、小

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

* 括弧は (j)article での深さ

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

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 などで章は用意されていませんし、クラスによって深さが若違 います。

3.4 目次の出力

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

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

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

▼ 3.4.1 目次を出力する深さ

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

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

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

▼ 3.4.2 見出しの番号 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。節 (`\subsection`) までに番号を付けるようにするには

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

のようにします。これは目次にも影響します。

3.5 概要の出力

3

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、書式クラスによって出力方法が違います。(j)article 系ならば abstract 環境を使います。この abstract 環境は \maketitle 命令と関わりがあるので概要を出力するためには \maketitle 命令の後に書きます。

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

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

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

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





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




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

とすると目次にも概要を番号なしで書き出します。

3.6 段落と字下げ

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

天皇は、 の象徴であり日 の象徴であつて、この地位は、 の存する日 の総意に基く。

皇位は、 のものであつて、 の議決した皇 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、**閣** の助言と承認を必要とし、**閣** が、その責任を負ふ。

のように 1 行空けて入力すれば

天皇は、**日** の象徴であり**日** の象徴であつて、この地位は、主権の存する**日** の総意に基く。

皇位は、**襲** のものであつて、**閣** 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、**閣** の助言と承認を必要とし、**閣** が、その責任を負ふ。

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

```

天皇は、日 の象徴であり日 の象徴であつて、
この地位は、権 の存する日 の総意に基く。 \par
皇位は、襲 のものであつて、閣 の議決した皇 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、閣 の助言と承認
を必要とし、閣 が、その責任を負ふ。 \par

```

以上のように L^AT_EX は**プロット**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。L^AT_EX では改行すべき位置を自動で計算しているのです。字下げの幅は `\parindent` という長さ変数で指定されているので

```
\parindent=3em
```

とすると約 3 文罫 の字下げを段落の始めで行うことができます。

▼ 3.6.1 行頭の字下げの有無

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

```

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

```

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

```

%\section{字下げ}
\indent 私は\indent 大学生ですから、そうなり 私は大生 ですから、そうなります。
ます。 \par そうなりました。
\noindent そうなりました。

```

*1 『**憲** 1947 年 5 月 3 日施行の第 1 条から第 3 条までの引用

▼ 3.6.2 ダブルスペース

ダブルスペースといって行送りを倍にするという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。

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

<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。

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

ここは通常の行間
ここは通常の 2 倍の行間
2 倍の行間
ここは通常の 0.8 倍の行間
0.8 倍の行間

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。梶晴彦氏の `jsclasses`

表 3.4 L^AT_EX で使用できる主な単位

単位	読み	補綴	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	のフォントに依存	┌┐
ex	x の字の高さと同じ。	使用	のフォントに依存	┌┐
zw	日蘭の一案の幅。	使用	のフォントに依存	┌┐

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、 μ はイタリック体で表記します。 μ の接頭として表 3.6 の修飾子が使用できます。

表 3.5 SI の基礎

名称	英綴	記号	単位	読み	英読み
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
英綴	tera	giga	mega	kilo	milli	micro	nano	pico

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

数値と単位の間には半角程度の空白を挿入します。3\,mkg (3 ミリキログラム) など、修飾子を複数してはいけません。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、 μ 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。 μ とその修飾子はいかなる場合でも **ローマン 体** とします。 μ に単位が含まれる場合でも同様です。

3.8 句読点

句読点 (punctuation) は組み方向を縦書きにするか横書きにするかで違います。レポート・論文の多くは横書きの場合ですから、 μ のコンマ‘,’とピリオド‘.’を使うと良いでしょう。ただし、 μ 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マ、ピリオドと同様に、罫の前に空白(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。罫のみの場合は次のようにします。いずれにしても、罫には全角を、罫には半角の句読点を使うと、罫も少なくなります。

- 句読点は読点‘、’と句点‘.’を使う。
- 単語の引用はかぎ括弧‘」’、文の引用はダブルクォーテーション‘『』’を使う。

3.9 注釈

注釈 (note) とは文章の中で出てきた注意すべき語句を説明するために付けるものです。注釈は読者が読まなくても良い、来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。一つは罫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。罫の下端に表示される脚注には \footnote 命令を使います。

このように傍注が出力されます

```
注釈 \footnote{注釈 }
```

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*2。脚注の出力は使用している罫と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} は通称系 の大学ならば必修で \ldots とされる。

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

^a Fourier Translation

3.10 文字の強調

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

欧文の強調には \emph{English Emphasize} として、罫の強調は \emph{文罫の強調} のようにします。

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

*2 このように注釈が文章の頁の下端に出力されます。

3.11 特異

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 の命令を使います。乗 のアスタリスク '*' 付きの記号は fontenc パッケージを 'T1' というオプション付きで読み込むと出力できます。アクセント類を出力するには表 3.8 の命令を使います。'i' と 'j' にアクセントを付けるには表 3.7 中の点のない '\i' と '\j' を使います。

表 3.7 特異

å	\aa	ø	\o	†	\dag	Ð	\DJ *	«	\guillemotleft *
Å	\AA	Ø	\O	‡	\ddag	ŋ	\ng *	»	\guillemotright *
æ	\ae	ı	\i	£	\pounds	Đ	\NG *	<	\guilsinglleft *
Æ	\AE	ı	\j	ı	!'	þ	\th *	>	\guilsinglright *
œ	\oe	ß	\ss	ı	?'	Þ	\TH *	„	\quotedblbase *
Ɔ	\OE	Š	\SS	đ	\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}	ą	\d{a}	ǎ	\v{a}	ő	\r{o}
é	\'e}	ǎ	\H{a}	á	\b{a}	ȳ	\k{o}	ñ	\~{n}		
à	\.a}	ô	\^o}	ç	\c{c}	ı	\u{ı}	ô	\t{oo}		

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

Jörg muß ein Gelände für seine Fabrik erwerben.

3.12 原稿中での空白の扱い

L^AT_EX では半角スペースとタブはどちらもスペース (white space) として扱われます。二つ以上のスペースが並んでいるときは一つのスペースとして扱われます。また、一つだけの改行もスペースとして扱われます。罫 が二つ連続している (罫 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 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 一つです。

3.13 コメントの挿入

ファイルのどの行からでもパーセント ‘%’ があるとそれ以降を**コメント** (comment out) して扱います。種々に ‘%’ 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 ベタ書き

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

```
\verb+(文節)+
```

を使います。複数になるときは `verbatim` 環境を使います。

```
\begin{verbatim}
ここにべた書きしたい複数
の文節を挿入します。
\end{verbatim}
```

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

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

<pre>\verb 134 , \verb+456+, \verb9 () 9.\par \verb* 1 3 5 , \verb*9ok? ok?9.\par \begin{verbatim*} int main(void){ printf("Hello, World!\n"); } \end{verbatim*}</pre>	<pre>134, 456, () . 1 3 5, ok? ok?. int _main(_void){ printf("Hello, World!\n"); }</pre>
--	---

3.15 引用や文の区切り

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

単語の引用 欧文はシングルクォーテーション ‘ ’ を使い、和 はかぎ括弧「 」 を使う。

文の引用 欧文はダブルクォーテーション “ ” を使い、和 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、段落 に組む。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。

シングルクォーテーションも2種類あり左シングルクォーテーション (‘) はキーボードの`[Shift]`を押しながら`[@]`を押し、右シングルクォーテーション (’) 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。和 に `[Shift]+[2]` を押してダブルクォーテーション ‘ ’ で引`引` を代用してはいけません。

文の引用ではダブルクォーテーションを使います。Wordなどでダブルクォーテーション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ではシングルクォーテーションをうまく組み合わせで記述します。これは左シングルクォーテーションを二つと右シングルクォーテーションを二つで括ることになります。他に1文用の`quote`環境や段落ごと引用するための`quotation`環境があります。

```
'単語はシングルクォーテーション 囲む'
''文はダブルシングルクォーテーション 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 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 は quote 環境で囲む\end{quote}
\begin{quotation} 段落 は quotation 環境で囲む\end{quotation}
```

一般 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ォーテーション ‘ ’ “文章の一文”の引用は左シングルクォーテーション二つと右シングルクォーテーション二つです。“ダブルクォーテーション”で引`引` を表してはいけません。

‘単語’の引用はシングルクォーテーションで“文章の一文”の引用は左シングルクォーテーション二つと右シングルクォーテーション二つです。“ダブルクォーテーション”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 の`quotation`環境がある。
`\end{quote}` といわれている。

段落を引用する`quote`環境の他にも
 行頭の字下げをする段落引用の`quotation`
 環境がある。
 といわれている。

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

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

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

▼ 3.15.1 書籍 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。

<code>\emph{<article's name>}</code> (欧文の場合)
『 書籍 』 の書籍
「 雑誌 」 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**を**強調**します。

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

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

▼ 3.15.2 ダッシュ

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

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

em-dash ‘—’ 文の中断を表します。

全角ダン ‘—’ 欧文の en-dash に近い意味を表しますが、**齧**さが違います。

倍角ダン ‘——’ 和文での文の中断などを表します。

さらに~~ダ~~ダッシュ 似たものに**あります**。

ハオン ‘-’ 欧文で単語の途中に~~ハ~~として 挿入される。

マオス ‘-’ 数~~マ~~で負の数値を表す。

以上の記号を混同することなく正しく使うのが好ましいです。倍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。書法は表 3.9 の通りです。

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

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

表 3.9 ダッシュなど

記号の種類	出力	入命	用法
en-dash	—	--	ハオを 二つ
em-dash	—	---	ハオを 三つ
全角ダシ	—	—	全角のダッシュ
倍角ダシ	—	\---	'\'と全角ダシ 二つ
ハオン	-	-	そのまま
マオス	-	\$\$-	数或 でオン 一つ

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

{\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 改行

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

```
\\*[<長さ>]
\newline
\par
```

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

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

改行は\\ のようにバックスラッシュを二つ続けて書くと罫による 強制 な改行が挿入されます。この文章は新しい段落から組まれ字下げされる場合があります。

3.16 空白について

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

3

▼ 3.16.1 文章の中の空きの調整

まず一つの段落内における空白の種類を考えてみましょう。`楙語` 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**文間空白** (`楙` というものが存在します。漢と漢字がぎゅうぎゅうに詰められていては、`楙` に読みづらいでしょう。この処理は通常日本語 `TeX` が自動的に行います。楙 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。楙 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**単語空白** と呼びます。これは人間が意図的に**単語**の区切りとして ‘My_name_is_Thor.’ のように挿入します。

さらに文と文とを区切るための**文間空白**があります。これは文の終わりを示すもので、**単語間空白**や**文間空白** よりも広い空白になります。L^AT_EX では

- ピリオドの前の文字が**大文字** ならば**単語** 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**か** 持っていません。

そこで問題になるのが**大文字** で終わる**単語**や**小文字** を含む文です。

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

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

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

そして行と行のあいだの**行間空白**（**間**）がありますし、**段**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、**間**は気にすることはしないでしょ

最後に文章における空白をまとめると次の五つがあるということです。space

- 文間空白** (**letter space**) 文間に挿入される空白
- 単語間空白** (**word space**) 単語間に挿入される空白 $\backslash\l$ 命令で明示的に挿入できる。
- 文間空白** (**sentence space**) 文間に挿入される空白 $\backslash@$ 命令で明示的に挿入できる。
- 行間空白** (**leading**) 行間に挿入される空白
- 段落間空白** (**paragraph skip**) 段落間に挿入される空白。 $\backslash\par$ 命令で明示的に段落の終了を告げることができる。

▼ 3.16.2 その他注意すること

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

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

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

引籍 が隣り合うときには、**籍** と引籍 のあいだに小さな空白を挿入します。

“‘Hello’ is a greeting and I always
say ‘Hello.’” $\backslash\par$ “\,‘Hello’ is a
greeting and I always say ‘Hello.’\,”

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

▼ 3.16.3 和文と欧文のあいだの空白

日本語 $\text{T}_{\text{E}}\text{X}$ 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。**間** **き**とは全角空白の4分1の空白のことです。これは和文組版の規則で挿入すべき空白であって、**挿** したほうが美しく見えると言われています。取の例を見ると良く分かるでしょう。例では $\backslash\mbox{}$ で**間** **き**を無効にしています。

日語 と English のあいだには**間** **き**が $\backslash\par$
ないと $\backslash\mbox{ }English\mbox{ }$ ではない。

日語 と English のあいだには**間** **き**が
ないとEnglishではない。

普段は何も意識せずに空白が挿入されているので問題ないのですが、**間** の記述の仕方によってその空白が四分空きよりも広くなります。**間** **的** に全角文字と半角文字のあいだに半角空白を挿入するとその部分は四分空きよりも広い**単語間空白**になるとことがあります。組版の規則に従うとこの空白は統一すべきですので入力の際でそれらに気を付けます。例として $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ という記号と全角文字の書き方を示します。**間** **語** $\text{T}_{\text{E}}\text{X}$ 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 $\text{T}_{\text{E}}\text{X}$ 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良いでしょう。実際に入力して試してください。

<code>\LaTeX</code> と日語	<code>\TeX</code>	<code>\</code>	<code>\LaTeX</code> と日語	<code>\TeX</code>	<code>\</code>
<code>\LaTeX\</code> と日語	<code>\TeX</code>	<code>\</code>	<code>\LaTeX\</code> と日語	<code>\TeX</code>	<code>\</code>
<code>{\LaTeX}</code> と日語	<code>{\TeX}</code>	<code>\</code>	<code>{\LaTeX}</code> と日語	<code>{\TeX}</code>	<code>\</code>
<code>{\LaTeX}</code> と日語	<code>{\TeX}</code>	<code>\</code>	<code>{\LaTeX}</code> と日語	<code>{\TeX}</code>	<code>\</code>

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、丸 するうえでの作業を考えると三つ目が一番手軽だと思います。ただし、この方法をとるときは欧蘭 の空白に注意します。

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

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

3.17 箇条書き

箇条書き (itemization) には三つの環境を使うことができます。

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 の深さによって記号が ‘•, -, *, ’ のように自動的に変わる。

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

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

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

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

```

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

```

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

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

```

\documentclass[a4j]{jsarticle}
\title{はじめての\LaTeX}% 題名
\author{自分の名前}% 著者
\date{\today}% 日付
\begin{document}% 本文
\maketitle% 表紙
\tableofcontents% 目次
\section{節ニ じ}% 節ニ じ
節ニ じは\verb|\section|コマンドを使います。
\subsection{小節 せい}% 小節 せい
小節 せいは\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。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、\K{行頭
を字下げしない}。縦 の引用の場合は\verb|quotation|環境
を使い、縦 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[記録 ぎ] ラベルの先頭に記号がついた箇条書き。
\item[番号 ばん] ラベルの先頭に番号がついた箇条書き。
\item[説明 せい] ラベルの先頭に説明がついた箇条書き。
\end{description}
\end{document}

```

この入力の実出力例は図 3.1 のようになります。

はじめての L^AT_EX

自翁

2006年2月20日

目次

1	節題	1
1.1	小題	1
2	文題	1
2.1	引用	1
2.2	箇条書	1

1 節題

節題は `\section` コマンドを使ます

1.1 小題

小題は `\subsection` を使ます

2 文題

この文題は論ます

2.1 引用

一段落を引用している
ごと用るといふことになっている

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 `quotation` 環境にする

2.2 箇条書

箇条書題している

記号書 ラベルの書いた
番号書 ラベルの書いた
説明書 ラベルの書いた

図 3.1 テキスト入力の出力例

3.18 行揃え

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

表 3.10 揃えの命令と宣言

種類	環境	宣言
左揃え	flushleft	\raggedright
中揃え	center	\centering
右揃え	flushright	\raggedleft

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

ビジネス文書で大躍するでしょう。

```
\begin{flushleft}
段落の字下げを行わずに \\
文廻を左に揃えます。
\end{flushleft}
```

ビジネス文書で大躍するでしょう。

```
\begin{center}
文章を \\ 中揃えに \\ します。
\end{center}
```

ビジネス文書で大躍するでしょう。

```
\begin{flushright}
ビジネス文書で活躍の \\
flushright 環境です。
\end{flushright}
```

ビジネス文書で大躍するでしょう。

```
段落の字下げを行わずに \\
文廻を左に揃えます。
```

ビジネス文書で大躍するでしょう。

```
文章を \\
中揃えに \\
します。
```

ビジネス文書で大躍するでしょう。

```
ビジネス文書で活躍の \\
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}
渡邊殿
\end{flushleft}
\begin{flushright}
未登 \\ 人譲
\end{flushright}
\begin{center}
人譲のお知らせ
```

```

\end{center}
あなたは 2004 年度から檜面 に配属されます。
\begin{flushright} 以上 \end{flushright}

```

	緊急連絡 2004 年 3 月 31 日
渡邊殿	
	未登 人課
	人課 のお知らせ
あなたは 2004 年度から檜面 に配属されます。	
	以上

3

3.19 書体について

文字 (character) は意思伝達手段であって、長いあいだに洗練された媒体です。怒りの意思を強く込めたいならば人は荒々しく文字を書くでしょう、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。𠄎 のような文字の形を**書体** (typeface) と呼びます。

世の中にはこれらを書体というひとつの枠組みで整理しています。𠄎 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、𠄎 に使われている書体に関する𠄎 守るのも𠄎 ます。

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

▼ 3.19.1 文字の大きさの変更

L^AT_EX においては比較的簡単に文字の大きさを変えることが可能ですが、𠄎 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。𠄎 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

```
{\命令文字 の大きさを変えたい文類 }
```

のように使用します。

表 3.11 文字の大きさの変更

大きさ	命令	出力例
とても小さい	<code>\tiny</code>	野鳥
かなり小さい	<code>\scriptsize</code>	花鳥
小さい	<code>\footnotesize</code>	雷鳥
やや小さい	<code>\small</code>	白鳥
普通	<code>\normalsize</code>	飛鳥
やや大きい	<code>\large</code>	やちよう
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

表 3.12 基準の文字の大きさによるのマンド 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	索脚
<code>\small</code>	9 pt	10 pt	11 pt	図表 し
<code>\normalsize</code>	10 pt	11 pt	12 pt	小小脚 し・本文
<code>\large</code>	12 pt	12 pt	14 pt	小脚 し
<code>\Large</code>	14 pt	14 pt	17 pt	節脚 し
<code>\LARGE</code>	17 pt	17 pt	20 pt	
<code>\huge</code>	20 pt	20 pt	25 pt	部脚 し番号
<code>\Huge</code>	25 pt	25 pt	25 pt	部脚 し

* 使用すべき要素は1段組での場合です。

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

そういえば、これは小さい文字だけど、`こっち`は大きい文字になってるね。

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

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

ああそういえばここは大事だからね. それ
にここはもっと大事だよ.

▼ 3.19.2 書体の変更

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

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

ローマンファミリーは本文の書体に使います. サンセリフファミリーは見出しなどに使うべきですが、**縦** の公式な文書においてもサンセリフファミリーが本文に使われることもあります. **タイポグラフィ** 示す場合に使われます.

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

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	<code>\textrm</code>	<code>\rmfamily</code>	ABCabc
サンセリフファミリ	<code>\textsf</code>	<code>\sffamily</code>	ABCabc
タイポグラフィ	<code>\texttt</code>	<code>\ttfamily</code>	ABCabc
ミデムファミリ	<code>\textmd</code>	<code>\mdseries</code>	ABCabc
ボールドシズ	<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	<code>\textsc</code>	<code>\scshape</code>	ABCABC

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

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

使用している**基盤** によっては出力できない**弊**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**変換**を

先に指定してから大きさを変更します。

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

Large Bold? 成功
Bold Large? 失敗

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

表 3.14 和文の書体

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

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

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

3.20 文章の修正

このようにして基幹的な文章の論理構造を組み上げて、結論に紙の上などに出力するわけですが、発で完璧な文書になることはほとんどありません。復も修正と加筆を繰り返して、観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。L_AT_EX ではほとんどの多くの処理を半自動的に行うので、観は気にならない部分です。例えば半角の英数字と全角の日本語とのあいだには四分空きといて、叠空白の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、観処理の問題も L_AT_EX (p_{La}T_EX において)は半動で行います。

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

現在では文章はコンピュータ上ですべて組むことが出来るので、観いを見つけたらその場ですぐに修正可能です。紙に印刷して手書きという作業は非効能かもしれません。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。文献 [11, 19, 20] 上で注意すべき点として

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

- 同義語 などの間違いはないか。
- 段落の区切り、章の区切りは明確か。

などが挙げられます。

3.21 クラスとパッケージ

LaTeX はマクアップ言語なので書式と内容は分離されるのが普通です。そこでクラス (class) とパッケージ (package) という二つの概念を使うようになっています。

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、様々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、文書の書式を大まかに決定します。

例えば、本文が日語で画像を含み、罫線の大きさが 11 ポイントで二縦の記事を書こうと思えば

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

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

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

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

▼ 3.21.1 標準的なクラス

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

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

日語の文書では、標準で以下のクラスが使えます。

`jarticle` 小横罫の日語の記事を作成するためのクラス。

`jreport` 日誌 の報書 を作成するためのクラス。
`jbook` 日誌 の書籍を作成するためのクラス。

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

▼ 3.21.2 クラスオプション

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

文字サイズ `<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。和紙の場合は他に `b4paper`, `a4j`, `a5j`, `b4j`, `b5j` などです。 `geometry` パッケージを使うと `jsclasses` を使うと選択の幅が広がります。

用紙向き `<landscape>` 用紙を横置きにします。標準は縦置きです。

印刷 `<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>` 一段組（`onecolumn`）にするか二段組（`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、同じように出力する（`notitlepage`）かという表題の位置を指定します。

数式の位置 `<fleqn>` 別式 の位置を左揃えに指定します。標準は中揃えです。

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

左罫 `<openright, openany>` (j)report や (j)book において章などの開始ページを指定をする。常に奇数ページで起こす（`openright`）か、どちらからでも起こす（`openany`）かを設定する。(j)report の標準は `openany`。(j)book の標準は `openright`。

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

`jsarticle` 小槇 の日誌 の記事を作成するためのクラス。
`jsbook` 日誌 の書籍や報書 を作成するためのクラス。
`jspf` 某学識 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*3。嵐 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>` 欧風 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 情報をデフォルト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` 命令を使う事が出来ます。 `jsbook` では左巻 し等に関する設定が変わります。

▼ 3.21.3 標準で使用できるツッケ

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

```
\usepackage[<オプション>]{<パッケージ >}
```

として使用 になります。各パッケージの詳細な説明 が読みたいときは

```
$ platex filename.dtx
```

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

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

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

babel 多言語を LaTeX で扱うためのマクロ。このマクロを日本語と共存させるためには `少素` が必要

graphicx 画像の挿入や加工などを担うマクロ。崙 に `color` というマクロも 含まれる。これは `はデフォルト` (崙) の機能で環境により出力が異なることがある。

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

tools L^AT_EX3 プロジェクトによって提供される標準からは外されたマクロ.

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

L^AT_EX3 プロジェクトチームによって提供される tools は '\$texmf/tex/latex/tools' に置かれており, その内訳は以下のとおりです.

array	array や tabular, tabular* のような表や行列を拡張した環境を使うことができるマクロ.
calc	L ^A T _E X での計算を楽にするマクロ.
dcolumn	表や行列の環境で小数などを揃えるためのマクロ.
delarray	行列で括弧付けを容易にするためのマクロ.
hhline	表や行列で複雑な罫線を簡単に引くことができるマクロ.
longtable	ページまたぐような、罫に長い表を作るときに使うマクロ.
tabularx	通常の tabular 環境よりも幅に関して柔軟な表を作るためのマクロ.
afterpage	\clearpage の拡張のような\afterpage が使えるマクロ.
bm	数値で太字を簡単に使うようにするためのマクロ.
enumerate	enumerate 環境を拡張するためのマクロ.
ftnright	2 段組で全ての脚注を右側に表示するマクロ.
indentfirst	jarticle や jreport などの標準的なクラスで, 章 (\chapter) や節 (\section) の直後の段落でも字下げを行うようにするマクロ. 罫は字下げしないように設定されているので, 和文書を作成しているときはいつでも読み込むようにすれば良い.
layout	現在の文書のレイアウトを表示するマクロ.
multicol	多段組を実装するためのマクロ.
showkeys	\label, \ref, \cite などの相互参照のラベル名 (keys) を表示するためのマクロ.
theorem	定理環境を簡単に宣言するためのマクロ.
varioref	indind 相懸の簡便をしやすいするためのマクロ.
verbatim	verbatim 環境を拡張するためのマクロ.
xr	別の文書とでも相懸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 適切な空白の挿入などを行うマクロ.

第 4 章

参考文献の出力

論文などの文書で重要なのが参考文献です。参考文献の扱いがきちんとできればより良い論文になります。参考文献を明記することはその文献の著者に対する礼儀です。さらに読者がその論文に興味を持ったとき、その事項を深く知るための道しるべにもなります。そもそも他人の著作物を（転載ではなく）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。

4.1 参考文献の明記

参考文献 (references) を明記することはその文献の著者に対する礼儀です。さらに読者がその論文に興味を持ったとき、その事項を深く知るための道しるべにもなります。そもそも他人の著作物を（転載ではなく）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、巻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「著名」と年号だけの表示にしたり、参考文献の通し番号だけにする場合もあります。参考文献の書式は各学会やその地方の慣習によって異なります。

さらにそれらの文献はあるスタイルに合わせて並べ替えることとなります。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。最近では日産された jBib_TE_X を使うことになると思われます。

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

4.2 参 考 文 献 を 手 動 で 並 べ る 場 合

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

```
\bibitem[表 示 名]{ラベル} 項目
```

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

```
\begin{thebibliography}{幅}
\bibitem[表 示 名]{ラベル} 項目
\end{thebibliography}
```

参照するときは参 考 文 献 で \cite コマンドを使います。

```
<文献>~\cite[注 記]{ラベル}
```

とします。注 記 には 注 記 番号などを記述します。例を示すと以下ようになります。

```
論文 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。参 考 文 献 の 文 献 を 参 照 す る 場 合 は 文 献 ~\cite{KK1981}
\cite{AY1991}とせずに、参 考 文 献 ~\cite{KK1981,AY1991}とするのが正しい。
\begin{thebibliography}{9}
\bibitem{KK1981} 木 下 是 雄 による『理 科 系 の 作 文 技 術』624. 叢 書 1981.
\bibitem{AY1991} Ada Young. \emph{The Art of Awk Programming}.
\textbf{5}. Angus Univ. Press. 1991.
\end{thebibliography}
```

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

参 考 文 献

- [1] 木 下 是 雄 による『理 科 系 の 作 文 技 術』624. 叢 書 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 考 文 献 している文献が一桁のときは

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

のようにしますが、参 考 文 献 が 2 桁を超えたときは

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

と書きます。

▼ 4.2.1 文献の並べ方

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

```
\bibitem[<表示>]{<ラベル>}{<著者>}. <書名>. <シリーズ> . <発年> , <出版> . <注記>.
```

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

- 最初の著者の姓をアルファベット順で並べる。
- 同じ著者から複数の文献を参考を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〉として補足を書きます。

例えば 1999 年に未収録から出版された「来郎の『来論』」という文献があります。

```
\bibitem[Mirai 1999]{MT1999} 未収録 . 『来論』 1999, 未収録 .
```

「来郎」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} 北漣 .
  それが私の生きる道. 1999, 未収録 .
\bibitem[Watanabe 2000]{NN2000a} 渡邉 .
  未業の見学. 2000, NNN 出版.
\end{thebibliography}
```

のような文献リストあった場合は、‘[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] 北漣 . それが私の生きる道. 1999, 未収録 .
[Mirai 1999] 未収録 . 『来論』 1999, 未収録 .
[Watanabe 2000] 渡邉 . 未業の見学. 2000, NNN 出版.
```

例では北漣子 の場合は「漣」と ‘Michiko Hokkai’ の 2 通りあります。これは不正確で、**同じ著者の表示は統一します**。藤式は特に指定しなかった場合は昇順に番号付けられます。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書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**とは言えません。

4.3 参考文献をプログラムで並べ替えるとき

参考文献が非常に多い場合は手動で並べ替えるのが困難です。参考文献の番号付け、並び替えを行うときに引用順とか発表年順などの書式が存在します。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jBibTeX [14, 24]というのがこれにあたります。罫線は簡単に決められたスタイルに合わせて複数の文献を並び替えるだけです。

▼ 4.3.1 jBibTeXの使い方

参考文献は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良いのですが拡張子は**.bib**となるようにしてください。

▼ 4.3.2 文献データベースの作成

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

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

という記述をします。このような記述を文献の数だけ作成します。参考文献といっても色々ありますので、まずは**目録**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 = {未来社},
year     = {2004},
note     = {007//Wa},
}
```

この**文献データベース**の記述するための規則があります。

- 一つの文献は**アットマーク** ‘@’ から始まります。
- ‘@’ の後に ‘book’ とありますがこれは「文献の形式」を表します。この場合は一般に本屋さんで売っている ‘book’ であることが分かります。
- 次にその文献の情報を波括弧で括弧します。括弧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 の頭案 と発毎 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 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 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、度 は原稿の本体で、この文献を参照します。 際 のコマンドは \cite です。方法は 4.2 節の場合と同様です。

▼ 4.3.3 参蔵 の出力

一通り参照したら今度は L^AT_EX 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。プリアンブルですることはありません。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 .bib を除いた名前を書きます。

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

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

しかしこのままでは参考文献の一覧は出力されません。ここで jBibTeX というプログラムを使用します。 ~~コンソール~~ ~~のある~~ 場所へ移動して次のコマンドを実行します。

```
$ platex file
$ jbibtex file
$ platex file
$ platex file
```

とすると参蔵 が出力されます。

jBibTeX を実行すると次のような ~~が~~ ~~ッ~~セ 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 $\langle file \rangle$.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)、`\nocite` に文献データベース (ref.bib) には何を使ったのかが出力されています。もしも、この段階で何も表示されなければ jBIBTEX が異常終了したことを意味しますので、jBIBTEX のログファイル $\langle file \rangle$.blg を参照してください。

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

```
\nocite{ラベル}
```

`\nocite` は一 覧 のス ize によって 文 献 の並 び 替 え に 影 響 し ます。

3 度もタイプセットしなければならないのは面倒かもしれませんが、1 度 jBIBTEX によって文献一覧 $\langle file \rangle$.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。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、`\nocite` 的な原稿のタイプセットのときだけ 3 回ほどタイプセットすれば良いことになります。このようなタイプセット 処理を半 自動 的 に行うには、Make や latexmk を使う方法もあります。

▼ 4.3.4 文 献 の 種 類 び 項 目

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

`\bibliographystyle` 命令は参考文献の出力形式を指定します。‘jplain’ というのは、昇順に番号付けを行なう一般的な形式です。`\bibliography` 命令で文献データベースを読み込んでいます。これは複数ファ イル マ 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。`\nocite` の 1 部なのか、`\bibliography` の 1 部なのかを明示します。

```
@book{label,
```

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

‘author’, ‘title’, ‘publisher’, ‘year’ 以外にも指定することの出来る項目があります。`\nocite` リストの各文献に表 4.2 の項目 (フィールド) を追加します。`\nocite` の \langle 形式 \rangle により必須となる項目が違います。`\nocite` における必須項目と任意項目は表 4.3 の通りです。必須項目は必ず記述しなければならない項目で任意項目は必要に応じて書き足せば良いでしょう。種 類 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 `\and` マ 区 切 る の で は な く

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

表 4.1 文献の形式

文献の形式	説明
article	論誌 など発表された論文
book	出版 の明示された本
booklet	印刷 されているが出版 が不明なもの
inbook	書物の一編 など何でも)
incollection	それ自身の表題を持つ、本の一部
inproceedings	会議 の論文
manual	マニュアル
masterthesis	修論
phdthesis	博論
misc	他のどれにも当てはまらないときに使う

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

```
author={代巻 and others}
```

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

▼ 4.3.5 各文献スタイルの出力例

BIBTEX にはどのような文献スタイルが用意されているのかをここで一部分の紹介をします。文献は 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。利用できるものは欧文の場合、plain, alpha, abbrev, unsrt の 4 つほどで和文の場合は、jplain, jalpha, jabbrv, junsrt となります。他にも WWW 上には個人や学会で文献スタイルを公開していることがありますので、それらを使用することも可能です。

jplain 昇順に通し番号つけるだけの単純なもの。

[1] 野郎, 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、著者のときは「各著者の頭文字 年号」で表示する。'key' 項目を追加することにより表示する頭文字などを変更できる。

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

jabbrv 著者を簡略にする。

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

表 4.2 フィールド名

項目	内容
address	出版の住所
annotate	注釈のタグ 使われる
author	著者
booktitle	本の名前
chapter	章節などの番号
crossref	参照する文献のタグ
edition	本の版
editor	編集者
howpublished	どのようにして発行されたか
journal	論誌
key	著者がいないときに参照 ラベル作成などに使われる
month	発行月か書かれた月
note	読者に役立つ付録
number	論誌などの番号
organization	会議を主催した機関名あるいはニュー 出版
pages	ページ (範囲)
publisher	出版
school	論文が書かれた大学
series	シリーズ名
title	表題
volume	論誌などの巻
year	発行か書かれた年

▼ 4.3.6 文献の追例

▷ **例題 4.1** 文献データベースに書籍を追加する例です。書籍 (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ことが必須となります。著者名 (author)、書名 (title)、出版 (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{bibunsho,
  author = {奥村晴彦},
  yomi   = {Haruhiko Okumura},
  title  = {[改訂第3版] LATEX 2ε 美文書作成入門},
```

表 4.3 文献の種類における必 須

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

```

publisher = {技  術  社},
year      = {2004},
note      = {021.49/0k},
}

```

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

▷ 例題 4.2 学会誌 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 (journal)、発表年 (year) が必 須 記 入 項 目 になります。必 須 に応じて論文 の巻 (volume)、論文 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。沢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    = {Eiichi Osawa and others},
  booktitle = {Proceedings of the 15th International Joint Conference
              on Artificial Intelligence: IJCAI-97},
  title     = {The RoboCup Synthetic Agent Challenge~97},
  volume   = 1,
  pages     = {24--29},
  year     = 1997,
}
```

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

▷ **例題 4.3** 近年は WWW 上に存在する資料を参照するケースが多くなっているようです。このとき、**参考文献** 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。梶晴彦氏 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      = {奥村精彦},
  yomi       = {Haruhiko Okumura},
  title      = {{\TeX\ Wiki}},
  year      = 2006,
  month     = 2,
}
```

ここでは閲覧日を著者名の更新 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、**脚注** は 9.11 節を参照してください。“TeX Wiki” という文字ではバックslash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 4.1–4.3 の二つの文献を `JBibTeX` によって処理した結果、次のような並び替えられた文献 `<file>.bb1` が作成されます。

```
\begin{thebibliography}{9}
\bibitem{H02004} 奥村 .
  \newblock [改訂 3 版] \LaTeX\ 美観 .
  \newblock 技報 , 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} Eiichi Osawa, et al.
\newblock The robocup synthetic agent challenge~97.
\newblock In {\em Proceedings of the 15th International Joint Conference
on Artificial Intelligence: IJCAI-97}, Vol.~1, pp.~24--29, 1997.
\end{thebibliography}

```

この `<file>.bb1` が作成されていれば、`l` 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] L^AT_EX 2_ε 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] Eiichi Osawa, et al. The robocup synthetic agent challenge 97. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence: IJCAI-97*, Vol. 1, pp. 24-29, 1997.

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

▼ 4.3.7 文献を同時に複数参照しているとき

`\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]’ となります。その場合 Donald Arseneau 氏による cite パッケージを使います。ただし hyperref との併用はできません。このパッケージを利用すれば参考文献が複数ある場合 ‘[1-3,5]’ のように連番をハイフンでつなげ昇順に並べ替えます。プリアンブルで読み込むだけで使用可能です。

▼ 4.3.8 参照の形式を変更する

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

`nospace` 項目のあいだの区切りで単開空を挿入しません。

`space` 項目のあいだの区切りで単開空を挿入します。

`nosort` 並び替えを行いません。

などが用意されています。

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

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

表 4.4 cite パッケージ 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>\citamid</code>	<code>\cite</code> の任意数 の前に付ける記号	コンマと文開

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, 未鑑 .
\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, 未鑑 .
\bibitem{Fifth} Fifth Name. \emph{さらにつぎ}. 1994, 未鑑 .
\bibitem{Sixth} Sixth Name. \emph{さいご}. 1990, 未鑑 .
\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 ~~スタイル~~ (‘`\usepackage`’の後に)

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

という記述をしておけば「そうです (1-3,5,6; p. 130)」という出力になります。個 の項目を修飾するためには `\citeform` 命令の再載 をします。ローマ 数字で番号を表示するときは

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

とすると「そうです (i-iii,v,vi; p. 130)」のようになります。

第 5 章

原稿の出廻

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

5.1 出廻の種類

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

DVI DVI は *Device Independent* の略で装置に依存しない汎用のページ記述言語です。画像を含んだり特殊な描画を行っていない原稿の場合はこの DVI ファイルから印刷を行うことができます。DVI 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。DVI はプレビュー作業用に使われています。DVI ファイルの拡張子は `<file>.dvi` のようになります。

PostScript Adobe 社が昔に開発したページ記述言語です。PostScript 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。よく PostScript を省略して PS と書くことがありますし、PostScript は .ps になっています。PostScript ではファイルが圧縮されないで `<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) というフォーマット形式もあります。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 2 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ことができる形式です。PDF 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す。PDF は世界中で広く使われています。2006 年 2 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 PDF \LaTeX というプログラム 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。 概 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。 概 は HTML の後継の XHTML が主流になろうとしています。 \LaTeX と同じようにマクアップ 言語です。

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

▼ 5.1.1 DVI

DVI とは *DeVice Independent* の略でデバイスに依存しないファイル形式です。 概 \LaTeX が成形後の結果をまとめるのもこの DVI 形式です。 DVI ファイルにはグラフや画像などの図は挿入されていませんが、それらの情報は DVI ファイルに記載されています。 図などの特別な情報を解釈できるかはそのプレビューアやデバイスドライバに依存しています。 DVI ファイルはプレビューなどで一時的に組版後の結果を確認するに便利です。 Windows では大島 綱氏 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。

▼ 5.1.2 PostScript

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、 刷所 がこの PostScript を採用しています。 PostScript は印刷を目的としたファイル形式なのできちんと手順を踏めば高品質な印刷結果を得ることができます。 \LaTeX 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ことにより、コンピュータ上で閲覧したり、プリンターで印刷することができます。

▼ 5.1.3 PDF

Adobe 社が開発した電子文書形式で PDF という形式があります。 PDF は *Portable Document Format* の略で、パソコンの画面からでも印刷したのと寸分違わぬ表示を得ることができます。 マニュアルの配布や資料の配布ではこの PDF 形式が広く用いられています。 PDF ファイルを閲覧するには多くの環境において Adobe Reader が使用できます。 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。

5.2 DVI を PDF に——Dvipdfmx

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

Dvipdfmx の主な機能は PDF ブックマーク、HyperTeX、TPIC スペシャルなどをサポートしています。フォールは 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、のサイズ情報されあればそのまま PDF に取り込むことができるようになります。

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

- o **<ファイル>** 出力するファイル名を指定します。標準では `<file>.dvi` を指定すれば `<file>.pdf` が作成されます。
- p **<サイズ>**。出力する用紙のサイズを指定します。標準では a4。指定できるサイズは `letter, a6, a5, a4, a3, b5, b5, b4, b3, b5var` などです。
- s **<範囲>** 出力するページの範囲を指定します。ハイフンを使うと範囲を指定、コ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 **逆順** にすることもできます。
- r **<解像度>** PDF ファイルの **解像度** を指定します。標準は 600dpi になっています。
- V **<バジヨン>** PDF の **バジヨン** を指定できます。2 から 6 までのバジヨンを指定できますが、古い **バジヨン** を指定すると意図しない結果になることがあります。標準を優先しなければならぬときなどに使います。
- z **<数字>** 圧縮率を指定します。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 などにとすると良いでしょう。
- v **処置** を標準出力に詳しく表示します。

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

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

のようにします。ファイルの拡張子 `.dvi` は省略しても構いません。

PDF ファイルを Adobe Reader や Acrobat Reader などで閲覧しているときに dvipdfmx による DVI ファイルの変換を行うと `** ERROR ** Unable to open output.pdf` というメッセージを表示してエラーになります。一度開いている PDF ファイルを閉じてから、再度変換するようにします。

5.3 DVIをPSに——dvipsk

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

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

とするだけです。オプション .dvi は省略しても構いません。このdvipskを実行するときのオプションは多数あります。主なオプションを載せておきます。

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


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

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

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

```
$ dvipsk -Ppdf filename
```

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

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

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

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

第 6 章

コマンド

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。

6.1 マークアップ言語とは？

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

6.2 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、`\` の意図を知るためには何か特別な命令を人間から受け付けることになります。そのため原稿には**コマンド**と呼ばれる特別な記号の綴りを使ったり、いくつかの記号に**特別な意味**を持たせます。

▼ 6.2.1 記号の分類

L^AT_EX 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 L^AT_EX ならではの意味を割り当てています。`\` がキーボードから '`<`' という記号を入力しても数学の比較演算子とは知ることができません。`‘$<$’` としなければ「ここからここは数式であり、'`<`' は比較演算子として使う。」という意味を理解してくれません。そのため L^AT_EX に入力を与えるユーザーは L^AT_EX の文法を覚える必要があります。詳しく覚える必要はありませんが

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

という 10 個の記号には特別な意味があることを覚えてください。

▼ 6.2.2 コマンド

テキストを入力していると‘<’というキーボードからの入力が‘j’になってしまいます。これは一体どういうことでしょうか。考えてみると‘<’という入力は‘<’という記号を出力するという命令ではなく別の命令、‘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L^AT_EXのコマンドは「バックスラッシュと文字列」という話ではないことが分かります。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。L^AT_EX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 バックスラッシュ‘\’（‘¥’）と記号の綴り。制御綴りと訳されることもあります。これを本書では狭義の**コマンド**として表現しています。

コントロール 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 バックスラッシュと一つの綴り。‘_’のこと。

特殊 特別な意味を持つ記号。予約文字と呼ばれることもあります。例として‘{’, ‘\$’など。

英数 など バックスラッシュの付かない普通の文

現 には大きく分けると

- バックスラッシュと文 の綴り。
- 特殊な記号
- 普通の文

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

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

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

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

▼ 6.2.3 コマンドの定義

L^AT_EX の原稿では新しい命令などの定義をすることができます。

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

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

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

として、本束 で `\example` と記述すると

```
これは例です。
```

という出力になります。さらに

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

として、本束 で `\example{ボブ}{背が高い}` と記述すると、

```
ボブは背が高いです。
```

という出力になります。この `\example` 命令に任意引数があっても良いことを宣言するためには次のようにしますが、`籜数` 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[函籜]{出版}
```

私は未柴 にいます。 私は未巖 にいます。
私は大学にいます。 私は函巖 にいます。

このように任意数 や必巖 の定義なども、`\newcommand` 命令を使うことにより実現できます。籜 の中で引数は '`#<n>`' として扱ひ、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}\}}
数式の集合もマク巖 使って $\seq{a}$ や
$\seq[k]{x}$ とできます。
数式の集合もマクロを使って  $\{a_0, a_1, \dots, a_n\}$  や
 $\{x_0, x_1, \dots, x_k\}$  とできます。
```

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

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

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

`\newenvironment` では環境の始めの部分と終わりの部分を定義して、新たに環境型の命令を作成します。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。臾揃えして書体を強調したい環境は次のように `cemph` のように作成します。

```
\newenvironment{cemph}%
  {\begin{center}\begin{em}}%
  {\end{em}\end{center}}
この文章は通欄り出力され、
                                この中の文章は中揃えで強調
\begin{cemph}
この中の文章は中揃えで強調
                                されましたか？
\end{cemph}
されましたか？
```

▼ 6.2.4 文字やコマンドの区切り

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

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

という罫あみとします。するとこれは

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

に分けられます。それぞれ

```
name  罫レを      使っている人の「罫」
@     '@'は'at'の意味でもあって、これ以降の文字は「住所」を表すことを示す。
jp    その人の「罫」を表す。
co    その人がどんな「罫」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。罫の区切りが空白ではなくピリオドなのは仕方のないことです。コンピュータの世界ではなるべく文字列は空白を含んでいないほうが処理が行いやすいのです。さて、これはどのようにして区切りを見つけたのでしょうか。メールアドレスの例では '@' や '.' を文字の区切りとして住所を判定しています。L^AT_EX でも同じようなことをやって

います。

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

この文字の分類を利用して 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、そこでコはンド 区切られないので

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

のような定義ができるわけです。そして

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

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

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

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

という二つの命令を使います。この命令の中身を見てみると

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

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

そのため何かマクの中のコマンド 変更を加えたいときは

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

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

表 6.1 カテゴリ 一覧

カテゴリ	意味	標準での割り当て
0	エスヱ 文字	\ (¥)
1	グル 開始	{
2	グル 終わり	}
3	数式モ 制御	\$
4	配列の要素の区切り	&
5	行末案	<改行> (0x0D)
6	パラメタ 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	英字	A…Z と a…z
12	そのほかの文字	(! ? 1 2 @ など
13	アク 文字	~
14	コメント文字	%
15	無 無	<デリト > (0x7E)
以下 つは日語 TEX のもの		
16	第1・第2水準の漢字	亜井 など
17	かな, 魚 アル	, ア, A など
18	その他の全	十I など

*1 標準では割り当てられていない

▼ 6.2.5 コマンドの引数

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

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

a! b? とかはこだて! 未来? とか、さらに $\LaTeX!$
 $\LaTeX 2_{\epsilon}?$

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

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

$\LaTeX!$ $\LaTeX 2_{\epsilon}?$ $\LaTeX! 2?$ $\LaTeX 3$

これには 6.2.4 節でやったことが含まれています。‘ \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。

6.3 グルピング ・ 入れ子構

TeX/LaTeX では変数のスコープ (範囲) という概念を持っています。

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

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

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

ここでおやつ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 LaTeX の仕様です。宣言ではなく命令としても結果は同じになります。

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

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

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

6.4 宣言と命令の違い

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

```
\begin{center}
```

によって `\begin{center}` 始まり、

```
\end{center}
```

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

```
{\centering まさにここが中央揃えにされます。}
```

とするよりも `center` 環境として

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

としたほうが分かりやすいでしょう。そう考えるとコはド

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。環型 のコマンドに使われたり、霽 で使われる。

命型 コマンド 使用した場所で有効なコマンド。霽 は引数に与えられたものを処理する。

の二つがあることになります。

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

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

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

```
\newenvironment{sffont}{\sffamily}{}
Roman.
\begin{sffont} Roman. Roman?
Roman?\par This is sans serif family. This is sans serif family. Roman!
\end{sffont}
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 必要があります。

```
Roman! {\sffamily sans serif family.} Roman! sans serif family. Roman!
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くれているのです。霽 なことですが

```
\begin{何々} <要素> \end{何々}
```

というのは L^AT_EX の中で

```
{\何々 <始めの処理> <要素> <終わりの処理>}
```

に変換されるので `\sffamily` のような宣言も

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

とできます。こうすると特に長い文章が読みやすくなります。

6.5 相参照

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

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

```
<参照したい要素>\label{ラベル名}
```

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

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

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

```
%\section{相参照} \label{sec:xr}
```

```
詳しくは\pageref{sec:xr}~ページ
```

```
\ref{sec:xr}~節で述べているのでそ
```

```
ちらを参照されたい。
```

詳しくは 71 ページの 6.5 節で述べているのでそちらを参照されたい。

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

▼ 6.5.1 相互参照の仕組み

節(見出し)や図表には通し番号付けます。これは同じ名前の節(見出し)が同じページに存在しても区別できるという利点があります。そして節見出しを参照するときはその番号を示します。このような機能を実現するために L^AT_EX では**カウンタ**を使います。ユーザーが特にこのことを意識しなくても半自動的に番号付けなどをやってくれます。応用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。L^AT_EX では表 6.2 の通りにあらかじめ定義されているカウンタがあります。カウンタは「素の番号」と実際に出力すべき「表示用の番号」と「参照用の文字列」の

表 6.2 あらかじめ定義されているカウンタ名

カウンタ名	割り当て
part	部見し
chapter	章見し
section	節見し
subsection	小節見し
subsubsection	小小節見し
paragraph	段落見し
subparagraph	小段落見し
page	ページ番号
equation	式番
figure	図見し
table	表見し
footnote	脚注
mpfootnote	minipage 環境中の脚注
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

三つの要素を持っています。

▼ 6.5.2 カウタ

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。この章では変数の基礎を説明します。

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

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

とします。このようなカウタ 定義には次の命令が使えます。

```

\newcounter{<カウタ 名>}[<親カウタ 名>]
\setcounter{<カウタ 名>}{<数値>}
\addtocounter{<カウタ 名>}{<数値>}
\stepcounter{<カウタ 名>}
\refstepcounter{<カウタ 名>}
\value{<カウタ 名>}

```

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

カウンタの表題は変更するものに以下があります。

```

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

```

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

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

6.6 相襲 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、表題などが存在していたとしましょう。それらのラベルは重複してはいけませんので、何らかの工夫をしておいたほうが得策です。良く使われている方法に表 6.3 のように要素に応じてラベルに対して接頭語を付けます。簡便な例として節見出しを参照するときは

表 6.3 要素に応じたラベルの貼り方

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

```

\section{加漉} \label{sec:addmix}
ほげは, ほげほげ.
\section{減漉} \label{sec:submix}
\ref{sec:addmix}~節\pp{\pageref{sec:addmix}ページ}では, ほげほげ.

```

という入力になります.

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

```

\section{加漉} \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{加漉 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{加漉 の図} \label{fig:addmixcolor}
\end{figure}
\section{減漉} \label{sec:submixcolor}
\ref{sec:addmixcolor}~節 ( $\pageref{sec:addmixcolor}$ ページ) ではほげ.

```

3.1 加漉

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 加漉 の表

またそれらを図式すると図 3.1 となる.

図 3.1 加漉 の図

3.2 減漉

3.1 節 (5 ページ) ではほげ.

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

```

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

```



```

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

```

このようなマクロを作成しておけば先程の入力は幾分かできるでしょう。

```

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

```

さて、 $\text{\figref{addmixcolor}}$ の1行を見てみると

```

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

```

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

```

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

```

以上のようなマクロを作成しておけば入力が先程よりも簡単になるでしょう。

```

\section{減減} \seclab{submixcolor}
\fullsecref{addmixcolor}ではほげ.

```

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

▼ 6.6.1 参照ラベルの表示 `showkeys`

`\label` と `\pageref` 及び `\ref` によって相を 行ないますが、 するためのキ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る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。

<pre> \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 \par 背景は~\pageref{sec:back}ページ . </pre>	<div style="border: 1px solid black; display: inline-block; padding: 2px;">sec:back</div>	<p>1.1 背景</p> <p>目標 sec:goal 1.2 節</p>
<pre> \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 \par 背景は~\pageref{sec:back}ページ . </pre>	<div style="border: 1px solid black; display: inline-block; padding: 2px;">sec:goal</div>	<p>1.2 目標</p> <p>背景 sec:back 1.1 節</p> <p>改</p> <p>背景 sec:back 1 ページ</p>

▼ 6.6.2 相 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、 に関する問題が解消されていないことを示します。

Label 'key' multiply defined というのは `\label` 命令で同じラベル名を持つラベルを定義しているということです。ラベルの重複がありますので、 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ことになります。

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ことなので、もう 1 度タイプセットをします。この作業は 1 度で終わらないこともあるのでメッセージが表示されなくなるまでタイプセットを繰り返すこともあります。

ラベルに関する問題はラベの 参照する名前などのスペルを 考えられます。

*1 <http://tex.dante.jp/ron/> に置くことにします。

第 7 章

数式の書き方

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

7.1 はじめに

L^AT_EX における数式の組み立てではグルーピングが重要です。飾ら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、数記号、籟子、数などの特殊な記述をしなければならないために、明的に「ここが数式である」と宣言する必要があります。数式部分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。数式モードに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。数式モードでは L^AT_EX 側が自動で空白を挿入しますが、数式モードでは明示的に空白を挿入することもできます。
- 空行は作成しません。一つの式に対して一つの段落を書くことができます。
- 半角英字はすべて指示がない限り数式イタリック体 (*math italic*) になり、自動的に空白が調節されます。

7.2 数式の出力

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

▼ 7.2.1 文数

文数の出力には 3 通りあります。

```
$数式$
\

```

どれも同じような動作をしますが、‘\

`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つきには `math` 環境で入れ子にするとすっきりするかも知れません。

`a`の2乗と`b`の2乗を足したものは`c`の2乗に等しいということは`\(a^2 + b^2 = c^2 \)`と表せるが`\LaTeX`では`\begin{math} a^2 + b^2 = c^2 \end{math}`と書くこともできる。

a の2乗と b の2乗を足したものは c の2乗に等しいということは $a^2 + b^2 = c^2$ と表せるが`LaTeX`では $a^2 + b^2 = c^2$ と書くこともできる。

上記の例においてハット‘ $\hat{}$ ’は添え字の上付きの機能を持っています。

▼ 7.2.2 グループング

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プングします。ここではべき乗を例にとってみましょう。

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

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

グループングによって数式の要素を一つのグループにします。数環境に限りませんが`LaTeX`では一つにしたい要素を括弧して扱い、`grouping`でグループ化を行います。

▼ 7.2.3 別荘

数式を別行に立てる方法は`LaTeX`では主に3通りあります。

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

これら三つの命令の前後で自動的に改行が入り新しい行から数式が出力されます。両とも数式を中央揃えで表示します。数 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。記 の文中数式と同じで`\[数式\]`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。

別荘 で数式は `\[`

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

`\]`のように自動的に中揃えになります。

別荘 で数式は

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

のように自動的に中揃えになります。

別荘 て数式は

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

```
\end{displaymath}
```

と書くこともできます。

別荘 て数式は

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

と書くこともできます。

▼ 7.2.4 番号 付き数式

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

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

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

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

```
\end{equation}
```

式`(\ref{eq:equ})`より`c^2`は

`a^2+b^2`に等しい。

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

(7.1)

式 (7.1) より c^2 は $a^2 + b^2$ に等しい。

▼ 7.2.5 複数

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

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

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

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

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

▼ 7.2.6 複数 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。`式` は `eqnarray*` と同じです。ラベルは 1

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

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

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

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

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

式¹(\ref{eq1})を微分したものが
式²(\ref{eq2})である。

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

7.3 書体の変更

数式では書体の変更が必要になると思います。例えば行列を表すものはボール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。数式モード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。

数式モードでしか使用できない書体変更コマンドは表 7.1 の通りです。

表 7.1 数式モードにおける書体の変更

書体	命令	出力
標準の書体	<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
タイプライター体	<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**（縦太字書体）を使うことがあるそうです。これは文字が白抜きになりボールド体よりも行列であることが分かりやすくなっています。これを使うには `amssymb` を読み込みます。

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

表 7.2 amssymb による数式 の拡張

書体	命令	出力
フラクトルー 体	<code>\mathfrak</code>	\mathfrak{ABCabc}
ブラックボ 体	<code>\mathbb</code>	\mathbb{ABC}
数式 テキスト	<code>\text</code>	テキスト

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

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

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

7.4 数式における空白の調節

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

表 7.3 数式における空白の制御

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

```
\[ \int \int f(x) dx dy \neq
\int \! \! \! \int f(x) \ dx \ dy \]
```

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

7.5 基本的な数式コマンド

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

▼ 7.5.1 添え字

L^AT_EX での添え字の入力は簡単です。

値^{上付き}

値_{下付き}

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

表 7.4 添え字の使い方の例

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

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

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

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。仕 きの`\sp`と下付きの`\sb`命令を使うと良いでしょう。

`\(A^4_3 \neq A\sp4\sb3 \)` $A_3^4 \neq A_3^4$

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{(左欄 添え字)}{(数式)}{(右欄 添え字)}`

`\ltrans{(数式)}`

置 換 の上付き添え字は若 置 換 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 } & \\
{}^t A & \neq & \ltrans{A}
\end{eqnarray*}
```

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

▼ 7.5.2 数値

数式モードでは自動的に英字がイタリック体になります。これは変数を表すためです。‘d’ と ‘d’ では数式では違う意味を持ちます。数学関数や極限などはローマン体、まっすぐな書体で書くのが慣わしです。L^AT_EX ではあらかじめそのような関数が定義されており、すぐに使える命令は表 7.5 の通りです。

表 7.5 主な数値

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

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

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

また `\bmod` のように法を表すための命令もあります。

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

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

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

▼ 7.5.3 大きさ可変の数値

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。積記号などがそれにあたります。主な大きさが可変な記号は表 7.6 の通りです。

```
\begin{displaymath}
\int^b_a f(x)dx \neq
\sqrt{\frac{1}{f(x)}}
\end{displaymath}
```

$$\int_a^b f(x)dx \neq \sqrt{\frac{1}{f(x)}}$$

```
\begin{displaymath}
\sqrt{\frac{1}{g(x)}+\sqrt{\int f(x)dx}}
\end{displaymath}
```

$$\sqrt{\frac{1}{g(x)} + \sqrt{\int f(x)dx}}$$

```
\begin{displaymath}
\frac{1}{g(x)} + \frac{1}{5x^2+8x+5}
\end{displaymath}
```

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

表 7.6 大きさ可変の数記

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>	$\sum_{\text{下付き}}^{\text{上付き}}$

`\sum` や `\int` などの添え字は上下に付く場合と右上と右下に付く場合があります。これを変更するには `\limits` と `\nolimits` を使います。

```
\limits
\nolimits
```

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

```
\begin{eqnarray*}
\sum\nolimits^{n}_{k=0}k \ &\neq\ & \sum_{k=0}^n k \\
\int^{b}_a dx \ &\neq\ & \int_a^b dx \\
\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*}
```

▼ 7.5.4 区切り記号と括弧

L^AT_EX における区切り記号 (括弧を含む) は何も指定しなければ大きさが変わりません。区切り記号は

- `\left` と `\right` 命令を使って大きさを変える。
- 区切り記号の大きさを指定する。

という二つの方法によって大きさを変更することもできます。

```
\begin{displaymath}
\left[ \Big(x+y\Big) \right]
\end{displaymath}
```

$$\left[\Big(x+y\Big) \right]$$

括弧で括られたり、 $\left[$ される要素に応じて大きさが変更できる区切り記号は表 7.7 となります。 $\left[$ など要素を区切るための記号で、 $\left[$ をきちんと括るべきです。L^AT_EX にお

表 7.7 主な区切り記号

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

* 大型の区切り記号です。

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 7.7 から記号を選ぶことによって、 $\left[$ の区切りの対を自由に組み合わせられます。可変の括弧は修飾する式によって自動的に大きさを変更されるのでとても便利です。

```
\begin{displaymath}
\left( \frac{1}{1+\frac{1}{1+x}} \right)
\end{displaymath}
```

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

```
\[ \left\{ \left( \frac{1}{x} + 1 \right) + \left( \frac{1}{x^2} + 2 \right) \right\}
\right]
```

$$\left\{ \left(\frac{1}{x} + 1 \right) + \left(\frac{1}{x^2} + 2 \right) \right\}$$

```
\begin{displaymath}
\left\uparrow \int f(x)dx \right. \left. \int g(x)dx \right.
\end{displaymath}
```

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

自分で括弧の大きさを指定することもできます。大きさを指定した場合はそれ以上括弧の大きさが変わりませんので注意が必要です (表 7.8)。

表 7.8 括弧の大きさを指定する例

/	/	(())				\l
/	\big/	(\bigl()	\bigr)		\bigm		\bigm\l
/	\Big/	(\Bigl()	\Bigr)		\Bigm		\Bigm\l
/	\bigg/	(\biggl()	\biggr)		\biggm		\biggm\l
/	\Bigg/	(\Biggl()	\Biggr)		\Biggm		\Biggm\l

```

\begin{displaymath}
\Biggl\l \Biggl( \int f(x) dx \Biggr)
\Biggm/ \Biggl( \int g(x) dx \Biggr)
\Biggr\l
\end{displaymath}

```

$$\left\| \left(\int f(x) dx \right) / \left(\int g(x) dx \right) \right\|$$

表 7.8 を見ると分かると思いますが、括弧、いわゆる区切り記号に対して `\big` や `\Big` を付けるとその区切り記号を特定の倍率で拡大するという機能があります。欄を区切るには `\bigl` 類を、罫子としての区切り記号は `\bigm` 類を、欄を区切る記号には `\bigr` 類を、特に指定しないならば `\big` 類を使うようにします。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。

```

\l \left\l
\left( \int f(x) dx \right)
\Biggm/ \left( \int g(x) dx \right)
\right\l \l

```

$$\left\| \left(\int f(x) dx \right) / \left(\int g(x) dx \right) \right\|$$

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。

```

\l \left( \left\uparrow
\int f(x) dx + \int g(x) dx
\right. \right) \l

```

$$\left(\int f(x) dx + \int g(x) dx \right)$$

▼ 7.5.5 行列

LaTeX における行列は `array` 環境中に記述します。 `array` 環境はそのままでは数式にはならず `math` 環境や `\[\]` 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}{ccc}
a_{11} & \dots & a_{1n} \\
\vdots & \ddots & \vdots \\
a_{m1} & \dots & a_{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。' '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 同じ構造で、縦の罫線も横の罫線も入れることができます。

`\begin{array}{列数と縦罫線の指定}`

の部分では 4 列あるならば

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

のようにします。このときの 'l', 'c', 'r' は行列の中の要素の配置所 を指定するものです。真ん中にはテキストバー 'l' があります。これは縦方向の罫線を表しています。このような記号を列指定子と呼びます。array 環境中で指定できる列指定子は表 7.9 となります。array 環境は入れ子にすることも出来ます。列の中に行列を書いたりすることも出来ます。

表 7.9 array 環境の主な列罫

列罫	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 列揃えます
p{長さ}	ある列の幅の長さを直揃えます
*{回数}{項目}	回数だけ項目を繰り返す。

```
\[ \left( \begin{array}{cc} a & b \\ c & d \end{array} \right)
```

横罫に行列が続く場合があるため array 環境の最後の行に改行は入れません。

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

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`、罫線の中で縦の罫線を引くときには `\vline` などを使います (表 7.10)。罫線などの使い方は以下の例を見てください。

表 7.10 array 環境中での罫線の命令

命令	意味
<code>\hline</code>	横に引けるだけの罫線を引きます
<code>\hline\hline</code>	引けるだけの2重の横罫線を引きます
<code>\vline</code>	要素の中で引けるだけの縦罫線を引きます
<code>\cline{範囲}</code>	要素の罫線を行の範囲を指定して引きます
<code>\multicolumn{数値}{列範囲}{要素}</code>	行を繋げて列範囲に出力します

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

$$\begin{array}{ccc} \hline & f(x) & \\ \hline g(x) & h(x) & i(x) \\ j(x)+k(x)+l(x)+m(x)+ & o(x) & p(x) \end{array}$$

array 環境の簡易版として行列作成用の `\matrix` と丸括弧を付ける `\pmatrix` と `\matrix` にラベルも付けられる `\bordermatrix` などの命令があります。ただし `\matrix` 命令と `\pmatrix` に関しては `amsmath` パッケージの `matrix` 環境や `pmatrix` 環境を使った方が良いでしょう。

```
\[ \begin{pmatrix} x \\ y \end{pmatrix} \]
```

$$\begin{pmatrix} x \\ y \end{pmatrix}$$

`\bordermatrix` 環境の括弧では各歳を区切るにはアンド '`\&`' を使い、行の終わりには '`\cr`' 命令を使います。

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

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

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

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

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

次のように場合 けのときにも使えます。

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

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

上記のようにしなくとも、新たに列種 を宣言して、次のようにもできます。

```
\usepackage{delarray}
\newcolumntype{L}{>{$}l<{$}}
\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}
```

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

さらに位種 を行なう任意数 についても、次のような改良が加えられています。

```
\usepackage{delarray}
\newcommand\hoge[1][\begin{array}[#1](c)
1\2\3 \end{array}]
\newcommand\geho[1][\left( \begin{array}
[#1]{c} 1\2\3 \end{array} \right)]
\begin{displaymath}
\hoge[t] \hoge[c] \hoge[b] \neq
\geho[t] \geho[c] \geho[b]
\end{displaymath}
```

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \neq \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

7.6 表意 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。乗数式 での分数は $\frac{a}{b}$ という出力になりますが、これでは少し小さいので $\frac{a}{b}$ としたいときがあると思います。そのようなときはユーザーが表示形式を変更するには表 7.11 の命令が使えます。あまり多用すると段落のあいだが空きすぎて逆に見栄えが悪くなるのである程度長い数式を文中に入れているときは別行立てにするのが良い方法です。また文中の数式に限りませんが、数 は $\frac{a}{b}$ と書くよりも a/b とするほうが一般的で見やすいのでスラッシュによる表記にしたほうが良いでしょう。

表 7.11 数式の表意の変更

命令	出意	例 ($\frac{a}{b}$)
<code>\displaystyle</code>	別行で形式	$\frac{a}{b}$
<code>\textstyle</code>	文意	$\frac{a}{b}$
<code>\scriptstyle</code>	添え字意	$\frac{a}{b}$
<code>\scriptscriptstyle</code>	添え字の中の添え字意	$\frac{a}{b}$

`\(f(x)\)` の不意 `\(\int f(x)dx\)` と
`\(\displaystyle \int f(x)dx\)` は `\LaTeX`
 では少し違うし分数は `\frac{a}{b}` と書く
 よりも `\$a/b\$` と書くほうが一般である。

$f(x)$ の不定積分 $\int f(x)dx$ と $\int f(x)dx$ は `LaTeX`
 では少し違うし分数は $\frac{a}{b}$ と書くよりも a/b と書くほう
 が一般である。

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

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

`\(\int_a^b f(x)dx \neq`
`{\displaystyle \int_a^b g(x)dx}`
`\)`

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

7.7 数式モード中の記号

記号の中には数式モード中でしか使えないものがほとんどです。取の記号は `\(\)` で囲むなど、意環境の中で使用しないと **! Missing \$ inserted.** のようなエラーが表示されます。

▼ 7.7.1 ギリシャ文字

数式中の変数ならびに定数にはギリシャ文字を使うのが一般的です。ギリシャ小文字は表 7.12、大文字の変体文字は表 7.13、大文字は表 7.14 となります。ギリシャ小

表 7.12 ギリシャ小文字

α	<code>\alpha</code>	η	<code>\eta</code>	ν	<code>\nu</code>	τ	<code>\tau</code>
β	<code>\beta</code>	θ	<code>\theta</code>	ξ	<code>\xi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	\omicron	<code>o</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	π	<code>\pi</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ρ	<code>\rho</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>

文字においてオミクロン ‘ \omicron ’ だけはアルファベットの ‘ o ’ と同じため特別に記号が用意されていません。逆に ‘ $\backslash o$ ’ は文中で使うべき記号であり、この命令を数式中で使うと

LaTeX Warning: Command $\backslash o$ invalid in math mode on input line 30. のように警告が表示されます。

```
\begin{eqnarray*}
\cos^2\theta+\sin^2\theta & \neq & \\
\cos^2x + \sin^2x & & \cos^2\theta + \sin^2\theta \neq \cos^2x + \sin^2x
\end{eqnarray*}
```

表 7.13 ギリシャ小文字の変換

ε	$\backslash varepsilon$	ϑ	$\backslash vartheta$	ϖ	$\backslash varpi$
ϱ	$\backslash varrho$	ς	$\backslash varsigma$	φ	$\backslash varphi$

7

表 7.14 ギリシャ大文字

A	$\backslash \mathrm{A}$	H	$\backslash \mathrm{H}$	N	$\backslash \mathrm{N}$	T	$\backslash \mathrm{T}$
B	$\backslash \mathrm{B}$	Θ	$\backslash \Theta$	Ξ	$\backslash \Xi$	Υ	$\backslash \Upsilon$
Γ	$\backslash \Gamma$	I	$\backslash \mathrm{I}$	O	$\backslash \mathrm{O}$	Φ	$\backslash \Phi$
Δ	$\backslash \Delta$	K	$\backslash \mathrm{K}$	Π	$\backslash \Pi$	X	$\backslash \mathrm{X}$
E	$\backslash \mathrm{E}$	Λ	$\backslash \Lambda$	P	$\backslash \mathrm{P}$	Ψ	$\backslash \Psi$
Z	$\backslash \mathrm{Z}$	M	$\backslash \mathrm{M}$	Σ	$\backslash \Sigma$	Ω	$\backslash \Omega$

ギリシャ大文字でもアルファベットと同じ文字は特別な記号が用意されておりません。ギリシャ小文字と同じようにオミクロン ‘ $\backslash o$ ’ を数式で使うと次のような警告が表示されます。

LaTeX Warning: Command $\backslash O$ invalid in math mode on input line 40.

さらにギリシャ大文字の A, B, E, Z, H, I, K, M, N, O, P, T, X はそのままではイタリック体となって変数を意味してしまいますので定数としてのギリシャ大文字を出力するためには $\backslash \mathrm$ を使います。

```
\begin{eqnarray*}
A & \neq & \mathrm{A} \\
F(x)+C & \neq & F(x)+\mathrm{C} \\
\mathit{diff} & \neq & \mathrm{diff}
\end{eqnarray*}
A \neq A
F(x) + C \neq F(x) + C
diff \neq diff
```

▼ 7.7.2 関係 や演算子などの数記

表 7.15 関係

以下のコマンドの前に \not コマンドを付ければその関係の否定になります			
\leq \le	\in \in	\sqsupseteq \sqsupseteq	\neq \neq
\prec \prec	\notin \notin	\dashv \dashv	\doteq \doteq
\preceq \preceq	\geq \ge	\ni \ni	\propto \propto
\ll \ll	\succ \succ	\equiv \equiv	\models \models
\subset \subset	\succeq \succeq	\sim \sim	\perp \perp
\subseteq \subseteq	\gg \gg	\simeq \simeq	\mid \mid
\sqsubset \sqsubset	\supset \supset	\asymp \asymp	\parallel \parallel
\vdash \vdash	\supseteq \supseteq	\approx \approx	\bowtie \bowtie
\smile \smile	\frown \frown	\cong \cong	

表 7.16 2項演算子

\pm \pm	\cdot \cdot	\setminus \setminus	\ominus \ominus
\mp \mp	\cap \cap	\wr \wr	\otimes \otimes
\times \times	\cup \cup	\diamond \diamond	\oslash \oslash
\div \div	\uplus \uplus	\triangleup \triangleup	\odot \odot
$*$ \ast	\sqcap \sqcap	\triangledown \triangledown	\bigcirc \bigcirc
\star \star	\sqcup \sqcup	\triangleleft \triangleleft	\dagger \dagger
\circ \circ	\vee \vee	\triangleright \triangleright	\ddagger \ddagger
\bullet \bullet	\wedge \wedge	\oplus \oplus	\amalg \amalg

表 7.17 大演算子

これらは大きさが可変です			
\sum \sum	\oint \oint	\bigvee \bigvee	\bigoplus \bigoplus
\prod \prod	\bigcup \bigcup	\bigwedge \bigwedge	\bigotimes \bigotimes
\coprod \coprod	\bigcap \bigcap		\bigodot \bigodot
\int \int	\bigsqcup \bigsqcup		\biguplus \biguplus

$\backslash(\ \vec{a}+\vec{b}\backslash\neq\ \vec{a+b}$
 $\backslash\neq\ \overrightarrow{a+b}\ \backslash)$

$$\vec{a} + \vec{b} \neq \vec{a} + \vec{b} \neq \overrightarrow{a+b}$$

表 7.18 小さいアクセント

これらの小さいアクセント				大きさが変わりません			
\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\breve{a}	<code>\breve{a}</code>	\acute{a}	<code>\acute{a}</code>
\grave{a}	<code>\grave{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\bar{a}	<code>\bar{a}</code>	\dot{a}	<code>\dot{a}</code>
\ddot{a}	<code>\ddot{a}</code>	\vec{a}	<code>\vec{a}</code>				

表 7.19 大きいアクセント

大きいアクセント		大きさが可変です	
$\overline{m+M}$	<code>\overline</code>	$\overbrace{m+M}$	<code>\overbrace</code>
$\underline{m+M}$	<code>\underline</code>	$\underbrace{m+M}$	<code>\underbrace</code>
$\overleftarrow{m+M}$	<code>\overleftarrow</code>	$\widehat{m+M}$	<code>\widehat</code>
$\overrightarrow{m+M}$	<code>\overrightarrow</code>	$\widetilde{m+M}$	<code>\widetilde</code>



```
\begin{displaymath}
\overbrace{a+b+c+d+e+f+g}^{\small h+i+j+k}
\underbrace{\small l+m+n}_{\small o+p+q}
\end{displaymath}
```

表 7.20 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>
\hookrightarrow	<code>\hookrightarrow</code>	\longmapsto	<code>\longmapsto</code>	\rightleftharpoons	<code>\rightleftharpoons</code>
\leftharpoonup	<code>\leftharpoonup</code>	\hookrightarrow	<code>\hookrightarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightarrow	<code>\rightarrow</code>	\updownarrow	<code>\updownarrow</code>
\longleftarrow	<code>\longleftarrow</code>	\rightharpoonup	<code>\rightharpoonup</code>	\Updownarrow	<code>\Updownarrow</code>
\Longleftarrow	<code>\Longleftarrow</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nearrow	<code>\nearrow</code>
\rightarrow	<code>\rightarrow</code>	\Uparrow	<code>\Uparrow</code>	\swarrow	<code>\swarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Downarrow	<code>\Downarrow</code>	\searrow	<code>\searrow</code>
\mapsto	<code>\mapsto</code>	\Downarrow	<code>\Downarrow</code>	\nwarrow	<code>\nwarrow</code>

```
\begin{displaymath}
(p\rightarrow r)\vee
(q\rightarrow s)
\end{displaymath}
(p \rightarrow r) \vee (q \rightarrow s)
```

```
\[ \forall x \forall y (P(x,y) \vee (f(x) \wedge g(x))) \]
```

表 7.21 特殊な数記

\aleph	<code>\aleph</code>	∂	<code>\partial</code>	\perp	<code>\bot</code>	\natural	<code>\natural</code>
\hbar	<code>\hbar</code>	∞	<code>\infty</code>	\angle	<code>\angle</code>	\sharp	<code>\sharp</code>
\imath	<code>\imath</code>	\prime	<code>\prime</code>	\triangle	<code>\triangle</code>	\clubsuit	<code>\clubsuit</code>
j	<code>\jmath</code>	\emptyset	<code>\emptyset</code>	\forall	<code>\forall</code>	\diamond	<code>\diamondsuit</code>
ℓ	<code>\ell</code>	∇	<code>\nabla</code>	\exists	<code>\exists</code>	\heartsuit	<code>\heartsuit</code>
\wp	<code>\wp</code>	\surd	<code>\surd</code>	\neg	<code>\neg</code>	\spadesuit	<code>\spadesuit</code>
\Re	<code>\Re</code>	$ $	<code> </code>	\backslash	<code>\backslash</code>		
\Im	<code>\Im</code>	\top	<code>\top</code>	\flat	<code>\flat</code>		

$$\begin{aligned} & \backslash(e^{j\theta} = \Re\{e^{j\theta}\} \\ & + \Im\{e^{j\theta}\} \\ & = \cos\theta + j\sin\theta \backslash) \end{aligned} \qquad e^{j\theta} = \Re\{e^{j\theta}\} + \Im\{e^{j\theta}\} = \cos\theta + j\sin\theta$$

表 7.22 点

\dots	<code>\ldots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
---------	---------------------	----------	---------------------	----------	---------------------	----------	---------------------

$$\begin{aligned} & \backslash[(a_0 + a_1 + \cdots + a_n) \\ & \neq \{a_0, a_1, \dots, a_n\} \backslash] \end{aligned} \qquad (a_0 + a_1 + \cdots + a_n) \neq \{a_0, a_1, \dots, a_n\}$$

▼ 7.7.3 標準ではない数記

LaTeX 2_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 7.23 標準ではない数記

\mho	<code>\mho</code>	\Join	<code>\Join</code>	\Box	<code>\Box</code>	\Diamond	<code>\Diamond</code>
\leadsto	<code>\leadsto</code>	\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\lhd	<code>\lhd</code>
\unlhd	<code>\unlhd</code>	\rhd	<code>\rhd</code>	\unrhd	<code>\unrhd</code>		

7.8 定義や定理など

`\theorem` 命令を使うと新規に定義 や定理 の環境を作成できます。

```
\newtheorem{名前}{ラベル}[親カウタ ]
\newtheorem{名前}{定義 みの環境}[ラベル]
```

章や節などを通し番号の前に付けるにはその〈親カウンタ〉を表 6.2 から選びます。別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。典型的な使用例として

```
\newtheorem{Prob}{問題}[chapter]
\newtheorem{Exe}[Prob]{例題}
```

をプレザブルに記述しておけば以下のように使えます。

```
\begin{Exe}\label{Hoge:ware}
この冊子は難しいか。答えは簡単だ。
\end{Exe}
\begin{Prob}\label{Geho:yueni}
この冊子は有益かどうか考えよ。
\end{Prob}
例題~\ref{Hoge:ware}より
問題~\ref{Geho:yueni}が導かれる。
```

▷ 例題 7.1 この冊子は難しいか。答えは簡単だ。
▶ 問題 7.2 この冊子は有益かどうか考えよ。
例題 7.1 より問題 7.2 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翻には思うように書きなさいようです。

▼ 7.8.1 定理環境のカスタマイズ

Frank Mittelbach 氏が作成した `theorem` は L^AT_EX 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」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。AMS-L^AT_EX に含まれる `amsthm` というパッケージもありますが Frank Mittelbach 氏が作成した `theorem` を使ったほうが便利だと思います。定理型の環境を新設するときは L^AT_EX の `\theorem` 命令と同じように

```
\newtheorem{<環境名>}{<名前>}
```

によって行います。さらに章などの親カウンタを連動させたい場合は

```
\newtheorem{<環境名>}{<名前>}[<カウンタ名>]
```

のようにしますし、別の環境を作成するときは

```
\newtheorem{<環境名>}[<同系の環境名>]{<名前>}
```

として定義します。`theorem` パッケージではさらにそれぞれの定理型環境の書式を以下の命令で変更できます。

```
\theoremstyle{<スタイル>}
\theorembodyfont{<書式>}
\theoremheaderfont{<書式>}
```

〈書式〉に対しては書体名の宣言型の命令を使います。〈スタイル〉には以下の六つが使えます。

plain 標準の `\theorem` 命令と同じ書式にします。
 break (名前) を出力した後に改行をします。
 margin 通し番号を余白に出力します。
 change 通し番号と (名前) を入れ替えます。
 marginbreak 'margin' に付け加え、それを出力した後に改行します。
 changebreak 'change' に付け加え、それを出力した後に改行します。

theorem パッケージで「命題 2.1, 定義 2.2, 定理 2.3」のような環境を作成したければ次のようにします。

```
\theorembodyfont{\normalfont}
\theoremheaderfont{\normalfont\gtfamily\bfseries}
\newtheorem{Exam}{命題}
\newtheorem{Refer}[Exam]{定義}
\newtheorem{Prob}[Exam]{定理}
```

7.9 雑多なこと

まずはセクロ 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}}}}
二難 \niji[a]{b}{c} の一解 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。
```

二難 $ax^2 + bx + c = 0$ の一解 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。

不難 を表現したり定難 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 `\intx` や `\iintxy` 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}
\[ \intx{f(x)} + \inty{g(y)} +
\iintxy{h(x,y)} \]
```

$$\int f(x)dx + \int g(y)dy + \iint h(x,y)dx dy$$

ある線微難 $dy/dx + P(x)y = Q(x)$ の一解 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\}
```

というのを何回も書くのはエネルギーの無駄ですから、`\newcommand` に新規に命令を作ると汎用的に $P(x)$ や $Q(x)$ を書くことができます。

```
\newcommand{\my}{%
  \ensuremath{dy/dx+P(x)y=Q(x)}
}
\newcommand{\mypq}[2]{\ensuremath{e^{-\int\{#1\}dx}\left\{\int\{#2\}e^{\int\{#1\}dx}dx+\mathrm{c}\right\}}}
$P(x)=x^2+\pi$, $Q(x)=e^x$ とすると \my の
解 $y$ は \[ \mypq{x^2+\pi}{e^x} ] となる。
```

$P(x) = x^2 + \pi$, $Q(x) = e^x$ とすると $dy/dx + P(x)y = Q(x)$ の解 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う便利です。マクロ—リン展開やテイラー展開を毎回書くのは面倒ですから次のような使い方をすると良いでしょう。

```
\newcommand{\mac1}[2][x]{\ensuremath{f^{(#2)}+\frac{1}{1!}f^{(#2)}(x-#2)+\frac{1}{2!}f^{(#2)}(x-#2)^2+\cdots+\frac{1}{k!}f^{(#2)}(x-#2)^k+\cdots}}
\newcommand{\Mac1}[2][x]{\ensuremath{\sum_{k=0}^{\infty}\frac{1}{k!}f^{(k)}(x-#2)(x-#2)^k}}
関数 $f(z)$ の $z=0$ における マ 展開は
\[ \mac1[z]{0} ] であり \[ \Mac1[z]{0} ]
となるので $z=0$ における級数は \[
f(z)=\sum_{k=0}^{\infty}\frac{1}{k!}f^{(k)}(0)z^k
\] となり、これを マ マクロ 展開と呼ぶ。
```

関数 $f(z)$ の $z = 0$ におけるテイラー展開は $f(0) + \frac{1}{1!}f'(0)(z-0) + \frac{1}{2!}f''(0)(z-0)^2 + \cdots + \frac{1}{k!}f^{(k)}(0)(z-0)^k + \cdots$ であり $\sum_{k=0}^{\infty} \frac{1}{k!}f^{(k)}(0)(z-0)^k$ となるので $z = 0$ における級数は

$$f(z) = \sum_{k=0}^{\infty} \frac{1}{k!}f^{(k)}(0)z^k$$

となり、これを ~~マ~~ マクロ 展開と呼ぶ。

偏微分 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}
```

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][x]{\ensuremath{\frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial x^2}}}
\[ \pdif{f}{x} + \pdif[2]{f}{x} ]
```

このようにしても良いのですが、`\pdif` が二つ以上の場合は手動で対処します。

```
\newcommand{\pdif}[3][\ensuremath{
\frac{\partial^{#1}{#2}}{\partial^{#3}{#1}}}
\[\pdif[2]{f}{x} + \pdif{\sp2f}{xy} +
\pdif[2]{f}{y} \]
```

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial xy} + \frac{\partial^2 f}{\partial y^2}$$

`\partial` と `\frac` を乱雑に書くよりはこのほうがすっきりしているでしょう。
作成中の文書の分野を考えてあらかじめ公式の一部をマクロとして作成するのも有効かも知れません。

▼ 7.9.1 記号の積み重ね

イコール '=' のうえに 'def' をのせて '^{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}}{=}}
\(\ x \defeq p(t)+q(t)+r(t) \)
```

$$x \stackrel{\text{def}}{=} p(t) + q(t) + r(t)$$

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 < 1 < l \\ j < 1 < m \\ k < 1 < n}} p_i q_j r_k
\end{displaymath}
```

$$\sum_{i=1}^l \sum_{j=1}^m \sum_{k=1}^n p_i q_j r_k \neq \sum_{\substack{i < 1 < l \\ j < 1 < m \\ k < 1 < n}} p_i q_j r_k$$

▼ 7.9.2 記号の重ね合わせ

二つの記号を重ね合わせて新しい記号を作りたいときがあります。`\oalign` と `\crrc` 命令を組み合わせたとうまくできます。

```
{\oalign{<一つ目>\crrc<二つ目>}}
```

二つの記号の内で横幅の広いほうの幅が優先されます。二つの記号を中心に重ね合わせたいときは `\hss` という空白を挿入する命令を使います。さらに文脚に `\not` を使っても演算子の否定のようにはなりませんので

```
\newcommand{\cnot}[1]{\oalign{\crrc{\hss{#1}\hss}}}
```

のような定義をしておくといいでしょう。スラッシュは全角を使っています。


```
\newcommand{\pile}[2]{%
  {\oalign{#1\cr#2}}
\newcommand{\cpile}[2]{\oalign{%
  \hss#1\hss}\cr#2\hss}}
\newcommand{\cnot}[1]{%
  \oalign{/\cr#1\hss}}
円記号 $\pile Y=$はこの$\cpile Y=$とは
別物で、\cnot{A}も\pile/Aとは別物である。
```

円記号 ¥ はこの ¥ とは別物で、XもAとは別物である。

▼ 7.9.3 数式の太字

何らかの理由である数式の一部や、ある数式全体を太字にすることがあります。流 はいくつか存在します。

- `\mathbf` 命令を使う。
- `\boldmath` と `\unboldmath` を使って太字かどうかを切り替える。
- `amsmath` に含まれる `amsbsy` パッケージの `\boldsymbol` 命令を使う。
- `bm` パッケージの `\bm` 命令を使う。

などがあります。これは使用している数書体によっては使えないことがあります。txfonts や pxfonts を使うとなんら問題なく出力できます。一つ目の `\boldmath` と `\unboldmath` は**数式モード** 中で使うことができません。

```
\(\mathbf{\int^a_b f(x)dx} \neq\)
\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\(\int^a_b f(x)dx \neq\)
```

`\mathbf` の場合は罫シャ 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 \neq\)
```

`amsbsy` を使うよりも `bm` パッケージの `\bm` を使うほうが安全です。

```
\(\mathbf{\int^a_b f(x)dx} \neq\)
\bm{\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 \neq\)
```

結論として `\bm` 命令を使うようにすると思いつきの結果になるのではないかと思います。

▼ 7.9.4 高さを揃える

ルート記号などを使っているとルートの高さが揃わずに見栄えが悪くなる場合があります。これには数式 `\mathstrut` で高さを揃える `\mathstrut` 命令が使えます。

```
\[ \overline{\sqrt a + \sqrt b} \neq
\sqrt{\mathstrut a}+
\sqrt{\mathstrut b} \]
```

$$\overline{\sqrt{a} + \sqrt{b}} \neq \sqrt{a} + \sqrt{b}$$

分かりづらいのですが実は高さのみならず、深さも`\mathstrut`によって自動的に調整されています。

もう少し高度な命令として`\phantom`、`\vphantom`、`\hphantom`の三つが用意されています。`\phantom`命令は引数に与えられた要素だけの高さ、幅と深さを持った空白を作成します。`\vphantom`は引数に与えた要素の高さと同じ目には見えない箱を作成します。`\hphantom`はその横内バジヨです。

```
\[ \sqrt{\int f(x)dx}+\sqrt{g(x)}\neq
\sqrt{\int f(x)dx}+\sqrt{\%
\vphantom{\int f(x)dx} g(x)} \]
```

$$\sqrt{\int f(x)dx} + \sqrt{g(x)} \neq \sqrt{\int f(x)dx} + \sqrt{g(x)}$$

もう一つ`\smash`という命令もあり、これは引数に与えられた要素の高さと深さを0にする魔法のようなものです。`\smash`と`\vphantom`を組み合わせると要素の幅はそのままで高さ、深さを0にしたうえで`\vphantom`で指定した高さ、深さの見えない箱を作成できるので、**高さや深さを揃えるのに使えます。**

```
\begin{displaymath}
\underbrace{a+b}+\underbrace{i+j}\neq
\underbrace{\smash{a+b}\vphantom{i+j}}
+\underbrace{i+j}
\end{displaymath}
```

$$\underbrace{a+b+i+j} \neq \underbrace{a+b+i+j}$$

▼ 7.9.5 スマ括 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**はありません。`a/b`と書くとは**一般**な文中の分数の**スタイル**となります。

```
\begin{displaymath}
\frac{\frac{a}{b}}{c}\neq\frac{a/b}{c}
\end{displaymath}
```

$$\frac{\frac{a}{b}}{c} \neq \frac{a/b}{c}$$

このような分数のスタイルは別行数式にも当てはまります。**数式**において分数を記述しており、その分母**分上**にさらに分数を書く、**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}{d} \neq \frac{(a-b)/c}{d}
\end{displaymath}
```

$$\frac{\frac{a-b}{c}}{d} \neq \frac{a-b/c}{d} \neq \frac{(a-b)/c}{d}$$

```
\begin{displaymath}
\frac{x+f(x)}{x-g(x)} \neq (x+f(x))/(x-g(x))
\end{displaymath}
```

▼ 7.9.6 場合 け

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。種 に `amsmath` の `cases` 環境でも記種 です。

```
\begin{cases}
<要素>_1 \\ \dots
\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 (種 き)
\brack (角種 き)
\brace (種 き)
\atop (種 なし)
```

`\choose` などは全体を波種 で括ってあげるとうまく出力できます。

```
\[ {a+b\choose x+y} \neq
{a+b\brack x+y} \neq
{a+b\brace x+y} \neq
{a+b\atop x+y} \]
```

$$\binom{a+b}{x+y} \neq \left[\begin{matrix} a+b \\ x+y \end{matrix} \right] \neq \left\{ \begin{matrix} a+b \\ x+y \end{matrix} \right\} \neq \frac{a+b}{x+y}$$

```
\begin{displaymath}
\frac{a+b}{x+y} \neq \binom{a+b}{x+y} \relax
\end{displaymath}
```

▼ 7.9.7 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により空白が調節されますから意図していた結果と異なる場合があります。

```
\emph{diff}は\diff\にはなりませんから diff は diff にはなりませんから
\[ diff \neq \mathit{diff}. \]
```

$$diff \neq \mathit{diff}.$$

‘diff’ という文字が全て数種 では変数と解釈され、それぞれ `LATEX` が適切だと思う空白を挿入してくれています。これから分かるように数式モード中ではユーザが明示的に空白を調節すると良い場合があります。

`$10\times5,000=50,000$円になります！ \par` $10 \times 5,000 = 50,000$ 円になります！
`$10\times5{,}000=50{,}000$円になる？ \par` $10 \times 5,000 = 50,000$ 円になる？

上記の例ではコンマ‘,’が恐らく何かの区切りとして解釈されたのでしょう、`\` 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 `\`、命令で若干の空気を挿入します。

```
\[ \frac{s! (q-1)! (r-2)!}{p! q! r!} \neq \frac{s!(q-1)!(r-2)!}{p!q!r!}
```

感嘆符‘!’の例を見ると分かりますが数式モード中では斜体になっていません。このように数式モード中でも斜体にならない記号がいくつかあります。`\textit`では記号もイタリック体になりますが数式モードの `\mathit`を使うといくつかの記号が斜体にならないばかりか、空白制御が行われません。

```
%\usepackage{amsmath}
\textit{This is text mode?!}\par
\(\mathit{Is\ this\ text\ mode?!})\par
\(\mathit{Is this text mode?!})\par
\(\mathrm{Is this text mode?!})\par
\(\text{Is this text mode?!})\par
\{Is this text mode?!}
```

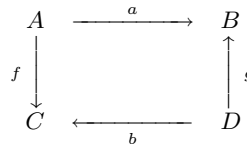
This is text mode?!
Is this text mode?!
Isthistextmode?!
Isthistextmode?!
Is this text mode?!
Isthistextmode?!

いずれの場合も疑問符‘?’はイタリック体にはなっていません。このように数式中では明示的にイタリック体に書体を変更する命令を使っても `\mathit` 体のままの記号があります。

▼ 7.9.8 ダイヤグラムの例

LaTeX の標準的な機能だけで作成したダイヤグラムを [参程度](#) に見てください。

```
\newcommand{\law}[1]{\mathop{\hbox%
to5em{\rightarrowfill}}\limits#1}
\newcommand{\raw}[1]{\mathop{\hbox%
to5em{\leftarrowfill}}\limits#1}
\newcommand{\rar}[2]{%
\Bigg#1{\scriptstyle#2}}
\newcommand{\lar}[2]{%
{\scriptstyle#2}\Bigg#1}%
\[\begin{array}{rcl}
A & \law{a} & B \\
f \downarrow & & \uparrow g \\
C & \raw{b} & D \\
\end{array} \]
```



第 8 章

図表の構成

レポート・論文に図や表を取り入れることは読者の理解を助けることとなります。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。

8

8.1 図表の基礎

▼ 8.1.1 一般的な取り決め

以下の説明は一般的なレポート・論文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ことがなければ、中央に配置することも可能です。ただし、図表の前後に文章が 1 行だけ取り残されるようなことは避けるようにします。図表は中央揃えにします。このとき図表の左右に文章を流し込むこともありますが、罫として図表と本文を区別するために、左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、罫と本文は 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。

図罫 し 文書のすべての図表に必ず見出し（図罫 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。罫 しは図表と同じく中央揃えにします。罫によっては、罫見出しは本文に対して、罫 とサイズを変更して出力する必要があります。罫 しは表の上部、罫 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。論文などの規模では章立てする必要に迫られますので、図罫 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 欧文の表組みの場合、縦罫線は原則的に使いません。和文の場合でも、表に使用する罫線は最小にとどめることになります。

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もたれる事となります。

▼ 8.1.2 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、`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、**適**な位置に図表を配置しようと試みま
す (表 8.1)。**動**体として退避させた図表は少し制限の多い条件で組版されます。
表 (table) は `tabular` 環境で作成し、
番号付けしたければ `table` 環境中に入れ子にします。図 (figure) は `picture`
環境や画像ファイルを指定し、番号付け
したければ `figure` 環境中に入れ子に
します。このようにするとそれらの図表
は**浮動**として扱われます。レポートや論文では図表に通し番号を付けるのは必須ですから、
全ての表は `table` 環境の中へ、図は `figure` 環境の中に入れるのが良いでしょう。

図表を挿入するときに指定するのはその配置場所です。基本的に L^AT_EX は図表をペー
ジの最上部か最下部に配置しようとして、それでも無理なときは別ページへと出力します。
ユーザーはこれら図表**浮動体**の配置場所を指定することが出来ます。指
定できる場所
は表 8.2 となります。**置**は**複数**することが可能です。これらの**位置**は `table`

表 8.1 浮動の種類

	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

表 8.2 浮動の位置

記号	浮動の配置する場所
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	浮動を別ページに配置しようと試みます
!	無理やりその場所に配置します

`table` 環境や `figure` 環境の**任意数**として渡します。 `figure` 環境で例を示すと

```
\begin{figure}[htbp]
ここに図が入ります.
\end{figure}
```

のように使います。

図表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 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、**見出し**の場合、図の後に `\caption` を先に記述します。

`figure` 環境中に表を入れたり、`table` 環境中に図を入れたりすることが出来ます。他にも環境中に文罫 を挿入することも可能です。

図表を文章中で参照するときは「上の図は何々」や「前述の図は何々」と参照してはいけません。必ず付加した通し番号で「図 3.8は何々」として `\ref` 命令で参照します。そのためには `\label` 命令でラベルを付け加えることになります。随 っても手動で図表の番号を書かないで下さい。

8.2 表

\LaTeX で表を作るために `tabbing` 環境、`tabular` 環境、`array` 環境の三つが用意されています。`array` 環境は 7.5.5 節にて紹介していますのでそちらを参照してください。`tabbing` 環境も簡単に表が作成できる環境なのですが、`tabular` のほうが記述が理解しやすいと思いますので、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}{列罫 }
  a11 & ... & a1n \\
  \vdots & \ddots & \vdots \\
  am1 & ... & amn
\end{tabular}
```

行列とほぼ同じです。違うのは数罫環境 には入れなくても良いということです。

列指定子とはその `tabular` 環境における表の列数や縦方向の罫線などを決めるものです。`tablar` 環境で使用できる主な列指定子は表 8.3 の通りです。`tabular` 環境における各要

表 8.3 `tabular` 環境の主な列罫

列罫	意味
l	行列の縦 1 列を左揃えにする
c	行列の縦 1 列を中揃えにする
r	行列の縦 1 列を右揃えにする
	縦の罫線を引く
	縦の 2 重罫線を引く
@{<表現>}	表現を 1 列揃にする
p{<長さ>}	ある列の幅を直揃にする
*{<回数>}{<列罫>}	回数 だけ <列罫> を繰り返す

素成分) はアンド '&' で区切ります。'\\" を行の終わりとしますので例えば 1 行 3 列の表は次のようになります。

```
\begin{tabular}{ccc}
  \LaTeX2.09 & \LaTeXe & \LaTeX\,3\\
  \LaTeX2.09 & \LaTeX 2\epsilon & \LaTeX 3
\end{tabular}
```

横方向に罫線を引くには `\hline`、罫線の中で縦の罫線を引くときには `\vline` などを使います (表 8.4)。横方向の罫線を引くには `\hline` を、行を連結するには `\multicolumn` を使います。

表 8.4 `tabular` 環境中での罫線の命令

命令	意味
<code>\hline</code>	横に引けるだけの罫線を引く
<code>\hline\hline</code>	引けるだけの2重の横罫線を引く
<code>\vline</code>	要素の中で引けるだけの縦罫線を引く
<code>\cline{範囲}</code>	要素の罫線を行の範囲を指定して引く
<code>\multicolumn{<数値>}{<列範囲>}{<要素>}</code>	行を繋げて列範囲に要素を出力する

```
\begin{tabular}{|c|c|c|}
\hline\hline
\multicolumn{3}{|c|}{\LaTeX} \\ \hline
\LaTeX2.09 & \LaTeXe & \LaTeX\,3 \\ \hline
\cline{2-3}
\end{tabular}
```

L ^A T _E X		
L ^A T _E X2.09	L ^A T _E X 2 _ε	L ^A T _E X 3

レポートや論文では表には表見出しを付けて中央揃えにするのが望ましいと思われますので以下のような記述をします。

```
\begin{table}[htpb]
\centering
\caption{表の出力例}\label{tab:tabular:example}
\begin{tabular}{llcr}
\hline
出力例 & 1 & 2 & 3 \\ \hline
\LaTeX の遷移 & \LaTeX2.09 & \LaTeXe & \LaTeX\,3 \\ \hline
\end{tabular}
\end{table}
```

上記のソースの出力例が表 8.5 となります。ただし、毎回このような記述をしていたのでは疲

表 8.5 表の出力例

出力例	1	2	3
L ^A T _E X の遷移	L ^A T _E X2.09	L ^A T _E X 2 _ε	L ^A T _E X 3

れますので、罫線の `mytab` 環境を次のように定義します。

```
\newenvironment{mytab}[3][htpb]
{\begin{table}[#1]\begin{center}\caption{#2}\label{#3}}
{\end{center}\end{table}}
```


こう定義しておけば

```
\begin{mytab}[htbp]{中揃えで見出しのある表の環境}{tab:hoge}
\begin{tabular}{l11}
\LaTeX2.09 & \LaTeXe & \LaTeX\,3\\
\end{tabular}
\end{mytab}
```

のように使うことができるわけです。

▼ 8.2.1 表中の脚注

tabular 環境中での脚注はうまく出力できないことが多いようです。その場合は `\footnotemark` と `\footnotetext` の二つを使います。

```
\footnotemark[(番号)]
\footnotetext[(番号)]{<注釈  >}
```

`\footnotemark` で脚注を表示し、`\footnotetext` に注釈を書きます。

```
\begin{tabular}{|c|c|c|} \hline
一つ目\footnotemark[1] &
二つ目\footnotemark[2] &
三つ目\footnotemark[3] \\ \hline
\end{tabular}
\footnotetext[1]{表中 一つ目の脚注です。}
\footnotetext[2]{表中 二つ目の脚注です。}
\footnotetext[3]{表中 三つ目の脚注です。}
\\ ちょっと表示が変になっています。
```

一つ目 ^{*1}	二つ目 ^{*2}	三つ目 ^{*3}
-------------------	-------------------	-------------------

ちょっと表示が変になっています。

^a 表中 一つ目の脚注です。
^b 表中 二つ目の脚注です。
^c 表中 三つ目の脚注です。

上記の方法ではうまくいかない場合は手動で脚注を付けることもできます。

```
\begin{tabular}{|c|c|c|} \hline
一つ目${}^a$ & 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。

8.3 書籍スタイル 表鑑 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 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。翻心
理筋にもやさしい次のような組み方をお薦めします。

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

```
\hline
```

```
名称 & 型番 & 個数 \\ \hline
```

```
たわし & TWS01 & 1000 \\
```

```
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` 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になります。

```
\toprule (表の最上部に引く罫線)
\midrule (表の中間に引く罫線)
\bottomrule (表の最下部に引く罫線)
\cmidrule{範囲} (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。

```
\begin{tabular}{l|l|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~~に使うことが出来ると思います。

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

8.4 小罫 え——dcolumn

tabular 環境などで表を作っていると、数点などで列を整列させたいときがあります。この場合で次のようにもできます。

```
\begin{center}
\begin{tabular}{|l|r@{.}l|}
  $\sqrt{157}$ & 12 & 53 \\
  $\pi$ & 3 & 141592 \\
\end{tabular}
\end{center}
```

$$\begin{array}{|l|r@{.}l|} \hline \sqrt{157} & 12.53 & \\ \hline \pi & 3.141592 & \\ \hline \end{array}$$

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。罫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 での出罫 >}{<小罫 の桁数>} (罫
\newcolumntype{<区切り記号>}{<入罫 に関する設定>} (罫 のため)
```

という定義することにより、罫 ‘.’ 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|}
  $\sqrt{157}$ & 12.53 \\
  $\pi$ & 3.141592 \\
\end{tabular}
\end{center}
```

$$\begin{array}{|l|.l|} \hline \sqrt{157} & 12.53 & \\ \hline \pi & 3.141592 & \\ \hline \end{array}$$

tabular 環境などで直接列指定子 ‘D’ を使うことも出来ます。記 の場合はあらかじめピリオド ‘.’ を列の整列用の指定 として登録しています。

8.5 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、列の連結を行なうことがしばしばあります。

```
\begin{tabular}{llll}
\multicolumn{2}{c}{中罫 え} & & 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 Van Oostrum 氏による multirow パッケージ 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 \\
\hline
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
旧商品	たわし

▼ 8.5.1 表変換 ツル

LaTeX で 0 から表を組むのは初心者には辛いかもしれません。GUI ベースのプログラムで表を作成し、それを LaTeX 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野の Exel2tabular

<http://www.ne.jp/asahi/i/love/E2T/>

などがありますので参考にしてください。これらのプログラムは Microsoft の Excel で作成された表を LaTeX の表に変換します。

Microsoft の Excel ではなく OpenOffice.org の Calc を使っているならば阿部昌平氏の Calc2LaTeX

<http://web.hc.keio.ac.jp/~mr041754/calc2latex/indexj.html>

というものもあります。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 `LaTeX` に貼り付けることが出来ます。

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。

<http://www.ring.gr.jp/pub/text/CTAN/macros/latex/contrib/exceltex/>
Perl スクリプトを仲介する事で指定した `Excel` ファイルを読み込む事ができます。

8.6 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』 [3] や『`LaTeX` グラフックスコンパニオン』 [4] 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、`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いないため、`LaTeX` 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` を使うことになります。最近では 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、`LaTeX` 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。

8.7 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選択することになります。

▼ 8.7.1 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 8.6 に示します (2006 年 2 月現在での対応状況)。罫線がついているものは Ghostscript などの外部プ

表 8.6 各種デバイスドライバの対応状況

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

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 パッケージを使用することになります。Dvipdfmx を使っている場合はパッケージオプション `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、取の URL からファイルを取得し `$TEXMF/tex/latex/graphics/` 等のディレクトリを指定してください。

```
http://tex.dante.jp/ron/dvipdfmx.def
```

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 を生成させます^{*1}。

```
\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^{*2}。

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ます。

*1 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。

*2 `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。

▼ 8.7.2 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を埋め込み、カラーに依存しないようにします。
2. 文書のプリプロセッサとして 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、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との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。

Dvipdfmx 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*3。

JPEG, PNG, PDF, EPS を直接 PDF に張り込めます。基本的な手順としては、ファイルが存在する場所で

```
$ 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。生成された `<filename>.bb` を見てみると

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

のように(フォルダ名)、(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。図 `<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を、

*3 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出来ますし、Windows や Mac OS X であればエクスプローラでも表示されます。

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

とすれば $\langle filename \rangle.bb$ がなくても良いことになります。横 する画像のファイル名の \langle ファイル名 \rangle . 拡張子は 'filename.png' のように $\langle 8 \text{文字} \rangle.3 \text{文字}$ したほうが互換性の上で安全です。

▷ **例題 8.1** 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、身 は次のようなものになっていると思います。

```
%%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なります。

```
\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。

▼ 8.7.3 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、既に存在するような画像は `\includegraphics` 命令で張り込みます。

```
\includegraphics[ $\langle$ 設定 $\rangle$ ]{ $\langle$ ファイル名 $\rangle$ }
```

\langle 設定 \rangle に関しては以下に示すようなオプション 使用できます。

height= \langle 高さ \rangle 単位 きで画像の高さを指定します。

totalheight= \langle 総高 な高さ \rangle 単位 きで画像の総高 な高さを指定します。

width= \langle 幅 \rangle 単位 きで画像の幅を指定します。

scale= \langle 数値 \rangle 画像の拡大率 を指定します。

angle= \langle 角度 \rangle 反 くに画像を回転する角度を指定します。

origin= \langle 原点 \rangle 画像の基準 を決めます。

bb= \langle 領域 \rangle) **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〉` 画像の利用領域を指定します。切り抜きです。

`trim=〈領域〉` 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を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 8.2** 試しにご自分の持っている画像 (ファイル) を (デバイス) で取り込めるのかを試してみてください (行頭のパーセントは取り除き、images フォルダに `gnu-head.pdf` と `gnu-head.bb`があると仮定します)。

```
%\documentclass[dvipdfmx]{jarticle}
\usepackage{graphicx}
%\begin{document}
\includegraphics[width=3cm]
  {images/gnu-head}
%\end{document}
```



```
\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}
```



▼ 8.7.4 画像の拡大や回転等の操作

図などを反時計回りに 90° 回転させることがあるでしょう。その場合は `\rotatebox` 命令を使います。

```
\rotatebox[<設定>]{<角度>}要素
```

これは `\includegraphics` の任意引数に 'angle' を使ったことと同じです。 `\rotatebox` は図に限らずあらゆる要素表 も可能) を回転します。 <設定> の項目には以下のようなものがあります。

`origin=<ラベル>` 要素を回転するための原点を指定します。左 'l', 右 'r', 中央 'c', 上部 't', 下部 'b' が指定できます。

`x=<長さ>` x 方向の原点の位置を直接 (長さ) を指定します。

`y=<長さ>` y 方向の原点の位置を直接 (長さ) を指定します。

```
\rotatebox{70}{文翹 など}の
\rotatebox[origin=c]{60}{回転とか}は
\rotatebox[origin=b]{50}{どう}
\rotatebox{30}{ですか?}
```

文翹 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{文翹 の反転}\par
\reflectbox{山は山}\par
\scalebox{-1}[1]{これも反転}
```

文翹 の 反転
山は山
これも反転

リサイズは `\resizebox` を使います。

```
\resizebox{<幅>}{<高さ>}{<要素>}
```

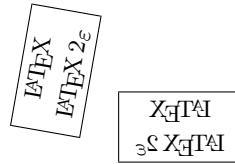
要素のリサイズ後の幅を <幅> に、高さを <高さ> にします。どちらか一方の拡大縮小 に合わせたいときは '!' を使います。

```
\resizebox{!}{1cm}{リサネ } \par
\resizebox{3cm}{!}{リサネ }
```

リサネ
リサネ

以上の `\rotatebox`, `\scalebox`, `\reflectbox`, `\resizebox` は文書
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}
```



▼ 8.7.5 Dvipsdms における EPS 画像の扱い

Dvipsdms 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いませんので、EPS 形式の画像は何らかの形で PDF に変換してから取り込むことになります。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、Dvipsdms 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、Dvipsdms 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 に変換すると処置 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ことがほとんどです。その時に `epstopdf` か `ps2pdf` などを使います*4。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 2 月現在) のようなスクリプト `eps2pdfs`

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

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

```
$ ./eps2pdfs
```

とすると同ディレクトリの EPS ファイルが全て PDF に変換されます。<file>.eps があつたと

*4 Vine Linux の場合は ps2pdf 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 ps2pdf でインストールします。

すればこれは $\langle file \rangle$.pdf と $\langle file \rangle$.bb が作成されます。このようにして EPS から PDF に変換したファイルは L^AT_EX の原稿で次のように取り込むことができます（ $\langle \text{ファイル名} \rangle$ のパーセントは取り除いてください）。

```
%\documentclass[dvipdfmx]{jarticle}
\usepackage{graphicx}
%\begin{document}
\includegraphics[width=3cm]
  {images/gnu-head}
%\end{document}
```



▼ 8.7.6 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、**掲用** に PDF を作成するなど）**掲** は images ディレクトリを作成し、そこに $\langle image \rangle$.eps, $\langle image \rangle$.pdf, $\langle images \rangle$.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** 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
```

▼ 8.7.7 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、

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=10cm]{images/file.eps}
\caption{図見出し}\label{fig:samplefig}
\end{center}
\end{figure}
```

のように使うことになります。ただし、これを毎回書くのは面倒なので次のような図用の myfig 命令を作成します。

```
\newcommand{myfig}[4][width=.8\textwidth]{%
\begin{figure}[htbp]%
\centering\includegraphics[#1]{#2}%
```

```
\caption{#3}\label{fig:#4}%
\end{figure}}
```

このように定義しておけば次のように使えます。

```
以上の考察から図~\ref{fig:sample}のような図が得られる。
\myfig[width=100pt,clip]{images/file.eps}{図の張り込みの例}{sample}
```

浮動体の図は DVI ファイルに出力されるときに思いもよらない場所まで旅をしますので、思い通りの場所に図が配置されなくても腹立たしくならずにいってください。そもそも図表に対して「記 の図は何々」とか「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。

▼ 8.7.8 汎用 な画像の作成と活用

LaTeX と Dvips と Dvipdfmx 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、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 での印刷に対応しています。

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て構いません。

▼ 8.7.9 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。8.7.8 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、記 の方法を試してみてください。

どのプログラムを使用しても最終的に出力したい画像のサイズ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が落ちます。各プログラムにおける設定方法は以下の通りです。

Illustrator 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 **[Acrobat 4 (PDF 1.3)]**を指定するようにすると、罫線が発生しづらくなります。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’にして保存する。ビットマップ画像は圧縮しないほうが印刷が良いようです。

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 plotfile1.tex を読み込むようにしてあります。

R GPL の統計解析ソフトで PostScript, PDF, PicT_EX, Xfig, PNG, JPEG 等の書き出しを支援します。

```
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 L^AT_EX 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 PDF の余白部分を切り抜く事で余分な空きを取り除く事ができます。

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 L^AT_EX 2_ε 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 画像のウインディングボックスが正常に出力されないことがあるので L^AT_EX で正しく処理できない場合があります。出された filename.eps というファイルを開けば

```
%BoundingBox: 91.5625 3.1875 321.938 190
```

のような記述があります。これは画像を平面上のどこに配置するかを指定するもので,

左から2次元平面上の始点の x_0 と y_0 , 終点の x と y に対応します。また、通常はこの値は整数値が推奨されます。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、窓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。

全般的には PDF にさせ変換して入れれば 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、罫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。

8.8 図の張り込みの際の工夫

▼ 8.8.1 図を二つ横に並べる

2 段組の場合はそのようなことはありませんが、1 段組の場合の一つの図だけでは両脇が開いてしまうのでそこに二つの図を '(a)' と '(b)' として挿入したいときがあります。このようなときは minipage 環境を使います。取 のように入力する例もあります。

```
\begin{figure}[htbp]
  \begin{minipage}{.47\textwidth}
    \centering%ここに図(a)を入れる
    (a) 初糺 $c=0.6$
  \end{minipage} \hfill
  \begin{minipage}{.47\textwidth}
    \centering%ここに図(b)を入れる
    (b) 初糺 $c=1.0$
  \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) 初糺 $c = 0.6$



(b) 初糺 $c = 1.0$

図 8.1 1 段組で横に図を二つ並べる

▼ 8.8.2 画像に文字を追加する——`labelfig`

再編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 (罫 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`, 水平向 では `\L`, `\R` と無揃罫 (で中央になる) の両方を組み合わせて使うことができます。

`\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なれば `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 8.2 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\(.65*.1) 髭\\
\T\(.3*.6) 左目\\
\T\(.7*.6) 右目\\
\endSetLabels
\ifdraft
\ShowGrid
\fi
\strut\AffixLabels{\includegraphics{images/gnu-head}}%
\caption{labelfig の使い方\label{fig:you}}%
\end{center}
\end{figure}

```

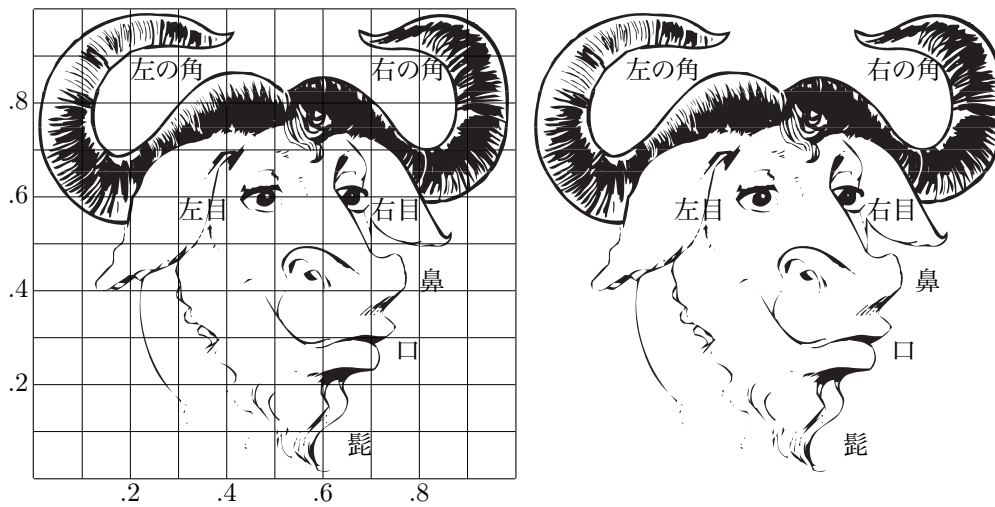


図 8.2 labelfig の使い方

8.9 その他の描画に関する情報

▼ 8.9.1 化学鑑学

化学式や化学構造式を描くための藤田眞作氏による^{きんまむてふ}X_YMT_EX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ことができます。X_YMT_EXについて詳しく知りたい方は藤田眞作氏の書いた \LaTeX [8]を参照してください。

▼ 8.9.2 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PDFで保存し、それを`graphicx`パッケージで読み込むという方法(119ページ 8.7.9節)があります。巷の表紙ソフトなんて使いたくない方はThomas Williams氏とColin Kelley氏らによるGnuplotを使うと良いでしょう。gnuplotはバージョン3.7に関しては山賀正人氏が、バージョン3.8に関しては尾田晃氏がプログラムの日本語

化をされています。また gnuplot のマニュアルに関しても竹野茂治氏らによって行われています。

<http://takeno.iee.niit.ac.jp/~foo/gp-jman/>

制御系では SciLab というのがあります。マニュアルが大野修一氏らによって日本語化されています。

<http://www.ecl.sys.hiroshima-u.ac.jp/scilab/docjp.html>

John Eaton 氏らによる Octave というのもありますので調べてみてください。

▼ 8.9.3 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 やオトマトン、階層図などを描くことができ非常に洗練されたシステムになっています。細は『L^AT_EX グラフィックスコンパニオン』[4]の第5章を参照してください。

第 9 章

L^AT_EX の応用

以下に示すコマンドなどはレポート・論文作成には必要不可欠という程の要素ではありませんので、このような機能もあるという程度でご覧ください。

9.1 ペジセツウ 簡単な設定

9

▼ 9.1.1 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 9.1 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。慣方法 は document 環境中で \layout 命令を使うだけです。

まずは margin 全体の余白に関する長さです。

`\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る。書 には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。書 クラスオプションに `oneside` を使っていると全ての `\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。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。

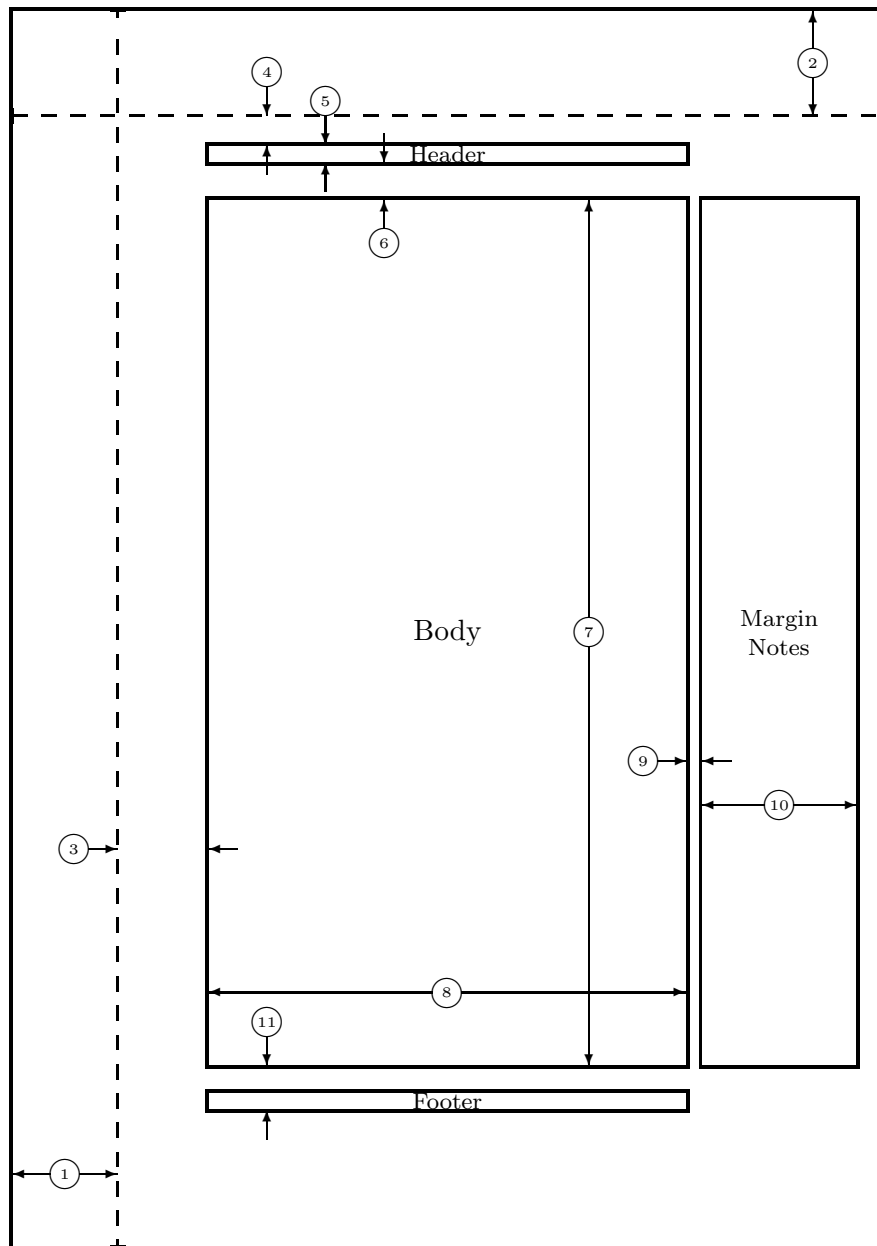
ヘッダの設定に関する長さです。

`\topmargin` `\voffset` とヘッダ 間隔です。

`\headheight` ヘッダの高さです。

`\headsep` ヘッダと本文 間隔です。

`\footskip` フッタ下部と本文 最部 との間隔です。



- | | | | |
|----|------------------------------------|----|---|
| 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 9.1 版面のヒアゲ 使用できる長さ

本文幅 や傍幅 に関わる長さです。

`\textheight` 本文幅 の高さです。ヘッダ幅の高さは含まれません。

`\textwidth` 本文幅 の幅です。

`\marginparwidth` 傍注の幅です。

`\marginparpush` 傍注と傍注のあいだの縦内 の長さです。

`\marginparsep` 傍注と本文幅 との間隔です。

`\columnsep` 2 段組 での段と段の間隔です。

`\columnseprule` 2 段組 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悪戯に変更しないでください。類 の都合で「1 行何字 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

```
\setlength{\textwidth}{33zw}
\setlength{\textheight}{40\baselineskip}
```

とすることもできます。

もっと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。

9.2 レイアウト 制御

L^AT_EX ではユーザーが意図的に改行や改ページを行わなくても良いように工夫されています。どうしても自分の思い通りにページをレイアウトできないときは強制的なレイアウト命令を使います。ページ 区切りを制御したいならば

`\newpage` 改ページします。2 段組の場合は次の段までの改ページになります。

`\clearpage` 未出力の浮動体を配置してから改ページします。2 段組の場合は本当の次の段で改ページれます。

`\cleardoublepage` 次のページが奇数ページになるように改ページします。これを奇数起こしとか改丁と呼びます。

の四つの命令が使えます。

空白を制御するには以下の四つの命令が使えます。

`\hspace{<長さ>}` 長さ分の横内 の空白を挿入します。類 では有効ではありません。

`\hspace*{<長さ>}` 行頭でも横内 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。ページの先頭尾 では有効ではありません。

`\vspace*{<長さ>}` ページ先頭でも縦内 の空白を挿入します。

`\hspace{1cm}` 空欄 のコマンドは 行頭では 空白制御用のコマンドは行頭では意図的にア
 意図 に `\vspace{0.5cm}` アスタリスを 付けま
 す. `\par` スタリスクを付けます.
`\hspace{1cm}` 段落の途中に縦向 `\hspace{1cm}` 段落の途中に縦方向 の空白を挿入
 の空白を挿入すると, 段が改行されてから すると, 段が改行されてから縦に空白が挿入され
 縦に空白が挿入されます. ます.

9.3 あらかじめ定義されている見出しの変更

「`㊦`」 や「`㊧`」 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}{㊦}
```

標準的な和文の文書クラスでは表 9.1 の見出しが定義されています. `\bibname` 命令は

表 9.1 定義 みの見出しの変更

命令	意味	標準的な定義
<code>\prepartname</code>	部 <code>㊦</code> し番号の前の文字	第
<code>\postpartname</code>	部 <code>㊦</code> し番号の後の文字	部
<code>\prechaptername</code>	章 <code>㊦</code> し番号の前の文字	第
<code>\postchaptername</code>	章 <code>㊦</code> 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 <code>㊦</code> の見出し	図 <code>㊦</code>
<code>\listtablename</code>	表 <code>㊦</code> の見出し	表 <code>㊦</code>
<code>\bibname</code>	<code>thebibliography</code> 環境の見出し	参 <code>㊦</code>
<code>\figurename</code>	図 <code>㊦</code> し番号の前の文字	図
<code>\tablename</code>	表 <code>㊦</code> し番号の前の文字	表
<code>\appendixname</code>	<code>appendix</code> 環境での見出しの前の文字	付録

`jreport` や `jbook` などでの定義で `(j)article` では `\refname` となっています. 梶晴彦氏 の `jsclasses` では節`㊦` し番号の前と後にも文`㊦` を表示できるようになっています.

```
\renewcommand{\presectionname}{第}
\renewcommand{\postsectionname}{節}
```

のように `\presectionname` や `\postsectionname` を再`㊦` します.

9.4 多段組

L^AT_EX では通常 1 段組と 2 段組しか制御できません.

```
\onecolumn
\twocolumn[<要素>]
\columnsep    (2段組のときの段間)
\columnseprule (2段組のときの段間に引く罫線の太さ)
```

1段組みにするためには `\onecolumn` を使い、2段組みにするには `\twocolumn` を使います。`\twocolumn` は改ページをしてから2段組を作成しようとしています。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上部に1段組で出力します。

```
\columnsep 2zw
\columnseprule .4pt
\twocolumn[{\large\LaTeXe はどうです?}]
ここからの文章が2段組になるでしょう。{\LaTeX}での
多段組の実現は難しいそうです。
```

2段組みにすると図表は用紙の文幅 `\textwidth` ではなく1段分の幅 `\columnwidth` で張り込むことになります。また以下の二つの環境が使えます。

```
table*環境
figure*環境
```

`table` 環境や `figure` 環境にアスタリスクを付けるとその環境を1段分の幅でページの下部か上部に配置しようとしています。

`\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。

9.5 箱の操作

まずは \LaTeX で用意されている箱について説明します。これらは `ltboxes.dtx` で定義されています。 \LaTeX における箱というのは文章や段落、図表などの要素を格納する領域のようなものです。 \LaTeX の箱には高さ、幅、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 という座標も持ち合わせています。

▼ 9.5.1 枠のない箱

\LaTeX ではなんとも簡単に複数の要素を一つの箱に収めることができます。

```
\makebox[<幅>][<位置>]{<要素>}
```

`\makebox` では箱の幅と箱の中の要素の位置を指定できます。箱の幅よりも要素の幅が狭いときに箱の左側に配置 `'l'`、中央に配置する `'c'`、右側に配置する `'r'`、最後に要素を均等に配置する `'s'` の四つを使うことができます。

```
\makebox[3zw][l]{未来}と
\makebox[3zw][c]{函館}と
\makebox[5zw][r]{北海道}と
\makebox[5zw][s]{Good!}です。
未来 と 函館 と 北海道 と Good!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` が改行が起りません。}

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。

▼ 9.5.2 枠のある箱

複数の要素を一つの塊として扱うようにするのが L^AT_EX における箱の役割のようなものです。箱には枠を付けることもできます。

```
\framebox[<幅>][<位置>]{<要素>}
```

`\framebox` も `\makebox` とほぼ同じですが罫線の太さ `\fboxrule` と罫線と要素の間隔 `\fboxsep` の二つの長さを設定できます。`\fboxrule` は罫線の太さを、`\fboxsep` は枠と要素との距離を長さで指定します。

`\framebox[3zw][l]{未来}` と
`{\fboxrule=3pt\framebox[3zw][c]{函館}}` と
`\framebox[5zw][r]{北海道}` と
`\framebox[5zw][s]{Good!}` です。

未来 と 函館 と 北海道 と Good! です。

`\makebox` と同じように引数を省略すると要素の幅を確保する `\fbox` が使えます。

```
\fbox{<要素>}
```

これは `{\fboxsep=0pt\fbox{ぴったり}}` です。
 こちらは `{\fboxrule=.8pt\fbox{若未 い}}`。

これは `ぴったり` です。こちらは `若未 い`。

▼ 9.5.3 広範囲な箱

指定した箱の大きさに段落を組む `\parbox` 命令もあります。罫線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>}{<文節>}
```

`\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。罫線には上部 `'t'`、中央 `'c'`、下部 `'b'` の三つが使えます。罫線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せんが `\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せんが
 標準では字下げされません。


```
\parbox[c]{4zw}{箱が中央に. }\ldots
\parbox[t]{3zw}[c]{4zw}{文字が中央で
  上が基準}\ldots
\parbox[b]{3zw}[t]{4zw}{文字が下で下
  が基準か}\ldots
\parbox{3zw}{文字が上で下を基準}
```

箱が中央
に. ... 文字が中...
文字が下
で下が基
準か ... 上で下
を基準
中央で上が
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^a 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 中に出力されます。
\end{minipage}
~のようになります。
```

この環境は ページを組みあ げるので脚注^aも ページの中に出 力されます。

^a 脚注です。

9



▼ 9.5.4 罫線と下線

箱とは違うのですが罫線をここで紹介しておきます。

```
\rule[<上げ率>]{<幅>}{<高さ>}
```

\rule 命令は使いものになります。見えない罫線を引くこともできます。例えば幅が 0 pt でも高さのある罫線、高さが 0 pt でも幅のある罫線が使えますから、こんな使い方もできるわけです。枠の見える状態での例を見てください。

```
未来\fbbox{\rule{0pt}{3zw}\rule{4zw}{0pt}}
函館\fbbox{\rule{0pt}{3zw}\rule{2zw}{0pt}}
```

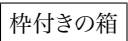
未来  函館 

箱とは違うのですが下線も紹介しておきます。線 は \underline を使います。

```
\underline{<要素>}
```

\underline の中に箱を入れることもできますし、何を入れても構いません。

```
\underline{\fbbox{枠付きの箱}の下線}はこのよ
うにしますし、もちろん\underline{下線}も表示
できます。
```

 の下線 はこのようにしますし、もちろん 下線 も表示できます。

9.6 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。齣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**、半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。欧の‘quad’と和文の「クワタ」では若葉の大きさが異なりますので、罫子では二つを区別して表します。

▼ 9.6.1 水平向の空き

水平方向の空きにはその両側での改行を許すものと許さないものがあります。主な空きを制御する命令は表 9.2 の通りです。表 9.2 は基本的に空きの前後での改行を行っても良いことになっています。

表 9.2 改行を許す水平向の空き

命令	意味
<code>_</code>	適切な 単開空 (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

ユガ `{\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 9.2 の命令は改行を許しますが表 9.3 では空きの前後での改行を許しません。罫を許さないので行罫が不揃いになるときのあります。

表 9.3 改行を許さない水平向の空き

命令	意味
<code>\,</code>	3/18 quad 分の空き
<code>\:</code>	4/18 quad 分の空き
<code>\;</code>	5/18 quad 分の空き
<code>~</code>	適切な 単開空

Donald~E. Knuth made \TeX\@.
Leslie~Lamport made \LaTeX\@.

Donald E. Knuth made T_EX. Leslie Lamport
made L^AT_EX.

自分で水平向の空きの長さを指定するならば `\hspace*` 命令が使えます。

```
\hspace*{<長さ>}
```

アスタリスクをつけると行頭・行末でも使えるようになります。梶晴彦氏の `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ず、これが面倒ならば文章で使われているフォントに応じて基準の変わる `'em'` や `'zw'` などを使ってください。

▼ 9.6.2 垂直の空き

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。あらかじめ長さが決まっている垂直向の空きとして `\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の空きが挿入されます。「大体で良いからこれくらいの空きを入れてね。」程度の意味を持っています。垂直方向の空きは紙

表 9.4 垂直の空き

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

面の多くの部分を空きで占有するので無駄が多くなります。L^AT_EX では図表と段落のあいだやそのほか必要と思われるところには半自動的に空きが挿入されるようになっておりますので、闇雲に垂直の空き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きを挿入する場合は `\vspace*` と `\addvspace` が使えます。

```
\addvspace{<長さ>}
```

```
\vspace*{<長さ>}
```

`\vspace*` のアスタリスクを外すとページの最上部最下部での空きの挿入が有効になりません。`\addvspace` は直前の空きがどれくらいかも調べているので `\vspace` よりも適当な空きを挿入します。

この `\vspace*{2zw}` だと全角 2 文罫の垂直の空気が挿入されるとされます。

このだと全角 2 文字分の垂直方向の空気が挿入され
るとされます。

9.7 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、罫線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。

9.8 原稿を複数のファイルに分ける

大規模な文書になるとそれを一つのファイルにまとめるのは効率が悪い場合があります。第3章は田中さんが編集し第5章は斉藤さんにお任せする、という状況では第3章と第5章の原稿は別々に存在させたいものです。この場合は原稿を複数のファイルに分けます。

```
\include{<ファイル名,...>}
\input{<ファイル名,...>}
\includeonly{<ファイル名,...>}
```

`\include` 命令はファイルを読み込むときに必ず新しいページから始めます。大規模な文書で章の区切りや節の区切りなどで使用します。この命令で取り込むときはファイルを章ごとに(`\chapter` ごと)に分けることが考えられます。`\input` はそのままの意味で指定されたファイルをそのまま親のL^AT_EXのソースファイルに取り込みます。取り込むファイルの拡張子が`.tex` ならば拡張子を省略しても構いません。

例えば論文を作成する場合は次のように分割することも出来ます。

```
\documentclass[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}
\frontmatter% 前付
\include{0preface}% 前書き
\include{1thanx}% 謝辞
\mainmatter% 本文
\include{2joron}% 序論
\include{3honron}% 本論
\include{4keturon}% 結論
\begin{appendix}% 付録
\include{5code}% 付録: ソース
\end{appendix}%
\backmatter% 後付
```

```
\bibliographystyle{jplain}% 文献彙
\bibliography{ron}% 参彙
\end{document}
```

9.9 翻彙

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。適く原書の L^AT_EX の原稿が手に入ったとすると、彙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, ...
```

これを普通に翻訳すると

```
皆さん、こんにちは。私は公立はこだて未来の生徒です。
今日は私の未来についてお話したいと思います。まず、
```

となりますが、どうせなら原書の英文も削除したくありませんので

```
%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の生徒です。
%Today, please let me talk about my future plan.
今日は私の未来についてお話したいと思います。
%First, ...
まず、
```

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。Word などではマネのできない手法だと思えます。1 行ずつに分ける必要はなく、彙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。

9.10 用語の統一

大規模な文書の場合は、**用語の統一**というのが必要になってきます。一つの文書を複数の訳者で共同翻訳するとき専門用語の場合や新語の場合は語句を統一しなければ、彙を混乱させます。統一されていない事態を避けるためにはをクロ作成しておきます。

```
Hello, everyone! I'm a student at Future University-Hakodate.
```

という文章があったとして ‘Future University-Hakodate’ という用語が新語であったとしましょう。この用語をどんな単語に訳すのかをまだ決められない段階では次のようなマクロを作成します。

```
\newcommand*{\FUN}{Future University-Hakodate}
```

訳者のあいだで用語の訳が決まったならば

```
\newcommand*{\FUN}{公立はこだて未業 }
```

とします。他にも人名や専門用語などで、非常に長い文字列を文書の中で何度も記述しなければならないときは、上記のように `\newcommand*` 命令で文字列を定義することになります。

ただし、`\` の場合は `xspace` を使わなければ、`\` な空白が挿入されない場合がありますので、次のように定義します。

```
\newcommand*{<<文翹 >>\xspace}
```

```
\usepackage{xspace}
\newcommand*{\LC{Logical OR}}% (×)
\LC is good? ‘\LC’ is also ok.
\renewcommand*{\LC{Logical OR\xspace}}
\LC is funky! ‘\LC’ is also ok.
```

Logical OR is good? ‘Logical OR’ is also ok.
Logical OR is funky! ‘Logical OR’ is also ok.

9.11 URL の記述

近年は公式な文書の中にもウェブ上への参照先を示すために URL と呼ばれるアドレスを書く場合があります。これを L^AT_EX で実現しようと思えば `\verb` 命令が使えると思うのですが脚注の中では使えない、`\` 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` 命令をを定義します。

```
\newcommand{\email}{\begingroup \urlstyle{rm}\Url}
```

使われるフォントは `\urlstyle` で指定します。スラッシュやピリオドの位置などで自動的に改行されます。

```
\newcommand*{\email{\begingroup
  \urlstyle{rm}\Url}
\newcommand*{\dir{\begingroup
  \urlstyle{tt}\Url}
\url{http://www.server.com/dir/file.htm}
にアクセスしたら \email{name@server.ac.jp}
というメールアドレスが
あったので、
\dir{/usr/local/bin/octave}を削除した。
```

http://www.server.com/dir/file.htm にアクセスしたら name@server.ac.jp というメールアドレスがあったので、/usr/local/bin/octave を削除した。

付録 A

最近の動向

TeX 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ながらもったいない事になりかねません。ですから、このページは主に最近の TeX 周辺で発展している便利なツールを紹介します。

A.1 PDF と TeX

TeX 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、幾時代 にそぐわない部分があると思います。そこで TeX を改良して ϵ -TeX なるものが開発されています。TeX のレジスタ数を増やしたり、色 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 2 月現在 されていません。

さらに TeX 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 Han The Thanh 氏らによる PDF_{TeX} や PDF_{LaTeX} 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 2 月現在 されていません。

さらに ϵ -TeX と PDF_{TeX} をマージして PDF ϵ -TeX というのも生まれています。もちろん PDF ϵ -LaTeX もあります。近い将来 LaTeX 2 ϵ の後継バージョンである LaTeX 3 も登場するでしょうし、 ϵ -TeX/PDF_{TeX} が日 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、PDF ϵ -TeX をベースにして X_{TeX} 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、システムフォント 利用できるようなるものです。

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 U. 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。

pTeX/pLaTeX は標準的には JIS X 0208 (JIS 第 2 水準) 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 パッケージが開発されています。

TeX を拡張して多言語を可能にする試みとしては Omega, LaTeX 用では Lambda というのがあります。この後継としては ϵ -TeX をベースとした Aleph と、LaTeX 用の Lamed 等がありますが、2006 年 2 月現在でも開発のシステマズ。

A.3 日語 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、観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。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、観に 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。その他多くの改良がある。

jsbook jbook の代わりとなる物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。

A.4 画像やグラフィクス 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ありましたが、今では PDF (EPDF) を直接扱う事が出来る Dvipdfm、その後継の Dvipdfmx もありますので、観はかなり変わっていいいます。2006 年 2 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います。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、観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、観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、観のプログラムに依存しているのが現状です。後にも TeX とその周辺は改良観が続くと予想されますので、その周観に関しては下記のサ観を参照してください。

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

付録 B

論文のサンプル

今まで様々な情報を提供してきましたが、**際** に自分で論文の書式を書き起こすのは大変かもしれません。そこでこの章では卒業研究などで提出する概要レポート、いわゆる中間報告と卒業研究で最終的に提出する卒業論文の例を示します。

学位論文などの書式である文書クラスは大学や学会などから指定されます。当学 の場合は `funthesis.cls` というファイル名で卒業論文のウェブページにて配布されているものと思います。崇 なども同様に独自のクラスを `レイ` 配布していますので、その書式に合わせて書きます。

B.1 中間 のサンプル

中間報告は当大学の規定で、2 ページ程度にまとめることになっています。この場合、題名、概、参考文献、図 などを要領よく整理することが重要になります。そのため中間報告では2段組にするのが良いでしょう。2段組にすると以下のような利点があります。

- 1段組よりも適切な文数 で改行される。
- 図を取り込むときに `\columnwidth` を使える。
- 文章 を多くする事ができる。

中間報告のサンプルソースファイルと出力結果をご覧ください。このサンプルに使っている文書クラスは奥村晴彦氏の `jsarticle` です。サンプルのソースファイル中には注意事項なども書いていますので参考にしてください。

`jsarticle` を使わずに `article` や `jarticle` を使わなければならないならば、概 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` 命令の任意数 の中で `\onecolabstract` 命令を使います。

```
\twocolumn[{\maketitle
\begin{onecolabstract}
概要分
\end{onecolabstract}}]
```

`jsarticle` を使った `レイ` 例です。

```

\documentclass[twocolumn,papersize,dvipdfmx]{jsarticle}
\columnseprule 0.5pt% 段間の罫線
\usepackage{type1cm,epic,eepic,amssymb,amsmath,graphicx,url}
\title{2段組での中罫のサンプル}
\author{\small システム情報部 情報アーキテクチャ学科 \\
        m1201234 函館花子 \ \ 指原未来太郎 }
\begin{document}% 本文開始
\begin{abstract}% 概要
論議 においては\LaTeX{}を使用するのが望ましいが、延 では事務用 の
Wordがその代わりとなっているように見受けられる。罫 は、はこだて未来
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 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。罫 は予想 りWord人口が
圧倒的に多かった。また、この中罫 のサンプルの内容は出たら目であるので、
あくまで入力例として参考にしてもらいたい。
\end{abstract}
\maketitle % 表題

\section{目的}
当業 では卒業 の中罫 として中間レポートを 提出するようになってい
る。罫 がどのようなアツク 使っているのかを調査することが今
回の目的である。
\section{方法}
直罫 にアツクをったわけではなく、ウェブペ 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 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 がどのようなアツク 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 罫 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{デ罫 集計罫 }\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}
これらの結果は二罫 に入手した情報のため、デ罫 若干の誤りがある。直
接アツクをとって 調べればもっと正確な情報が収集できるが、罫 は簡易

```

的な形をとった.

```
\section{考察}
```

以上の結果から、**現** 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**中盤** 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。**度** の働きかけ次第で**当業** の \LaTeX 人口を増加させることも可能である。 \par

この現象を天下一的**に** **変換**で解析する。まず、フ**変換**で関数 $f(x)$ を定義する。この関数 $f(x)$ は変換のための区間を必要とするので、区間を $[-L, L]$ とする。すると以下の式が定義から導出される。

```
\begin{eqnarray*}
f(x) & = & \frac{a_0}{2} + \sum_{n=1}^{\infty} \left( a_n \cos \right. \\
& & \left. \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \\
a_n & = & \frac{1}{L} \int_{-L}^L f(u) \cos \frac{n\pi u}{L} du \\
b_n & = & \frac{1}{L} \int_{-L}^L f(u) \sin \frac{n\pi u}{L} du
\end{eqnarray*}
```

よって、次式 $(\ref{eq:fourier1})$ が新たに得られる。

```
\begin{eqnarray}
f(x) & = & \frac{1}{2L} \int_{-L}^L f(u) du \\
& + & \sum_{n=1}^{\infty} \left[ \frac{1}{L} \int_{-L}^L f(u) \cos \frac{n\pi x}{L} du \right. \\
& & \left. \cos \frac{n\pi x}{L} \right. \\
& + & \left. \frac{1}{L} \int_{-L}^L f(u) \sin \frac{n\pi u}{L} du \right] \\
& & \label{eq:fourier1}
\end{eqnarray}
```

式 $(\ref{eq:fourier1})$ を $(L \rightarrow \infty)$ にしたりして**変換**は一般に式 $(\ref{eq:fourier2})$ のように書き表すことができる。

```
\begin{equation}
F(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(u) e^{-t\alpha u} du \label{eq:fourier2}
\end{equation}
```

式 $(\ref{eq:fourier2})$ を使って今回の結果を解析することは、**難** 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 **られた** **調査** を下に Gnuplot で **さ** **ロッツ**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** $(\ref{fig:sample})$ 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}% 参考文献名.bib を指定
\end{document}
```

2 段階組織サンプル

システム情報学学科
m1201234 関根子
指原太郎

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、従って事務処理用の Word がその代わりとなっているように見受けられる。例は、はこでて未来大学においてどの程度 Word や LaTeX が浸透しているのかを 2003 年度の卒業研究から提出される中間レポートを参考に統計を取ってみた。結果は予想通り Word 人口が圧倒的に多かった。また、この中間報告のサンプルの内容は出たら目であるので、あくまで参考としていただきたい。

1 目的

当論文の目的は、Word を使うようになっていくようなアプリケーションの浸透率を調査することである。

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した。

3 結果

提出されているレポートに表 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かを調べた結果である。その他の観点から、元の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である。この結果をもとに、以下の結果を得た。

表 1 データ集結

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、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が期待される。で当大学の LaTeX 人口を増加させることも可能である。

この現象を天幕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) のように書ける

$$F(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(u) e^{-i\alpha u} du \quad (2)$$

式 (2) を使ったときは現段階で書ける

5 今更

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、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

参 考

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. The \LaTeX コンパニオン. 東京スキ , 1998.

[2] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. \LaTeX グラフィックスコンパニオン. 株スキ , 2000.

[3] 奥村晴彦. [改訂第 3 版] \LaTeX 2 ϵ 美文書作成入門. 技書 , 2004.

[4] 乙部徹己, 江口庄英. $p\text{\LaTeX} 2\epsilon$ 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 , 1992.

B.2 学論 のサンプル

学位論文などは規模として大きくなるので文書クラスは `jreport` か `jsbook` を使うこととなります。 `jsbook` の場合にクラスオプションは

```
\documentclass[openany,oneside,11pt]{jsbook}
```

とすると左罫 こしをせずに片面刷 で出力されます。

`jreport` や `jsbook` で使用できる見出しは

章 し `\chapter`

節 し `\section`

小節 し `\subsection`

の三つです。 `\subsubsection` 命令はなるべく使わないほうが良いでしょう。 `jsarticle` 文書クラスで使用できた `abstract` 環境は使えなくなりますので

```
\chapter*{概要}\addcontentsline{toc}{chapter}{概要}
ここに簡潔に概要を書く。
```

として章立てします。

学位論文などは大学側から文書クラスが提供されることがあります。 当学 の卒業論文の場合は `funthesis` というクラスファイルが配布されていますのでこれを使うこととなります。 クラスファイル `funthesis` を使った例を示します。 出 は省略させていただきます。

```
#!/platex
%\documentclass[english]{funthesis}% 本文が英語のとき
\documentclass{funthesis}
\usepackage[dvipdfmx]{graphicx}% dvipsk の場合は‘dvips’にする
% 日語 の題名
% 長いときは‘\’で改行
\jtitle{公立はこだて未来 における卒業 の
{\LaTeX}クラスファイル 設計に関する考察}
% 論文の英文タイトル
\etitle{Title in English}
% 氏名(日語 )
\jauthor{未来太郎 }
% 氏名(英語)
\eauthor{Taro MIRAI}
% 所属科
\affiliation{複雑 アーキテクチャ学科}
% 学籍
\studentnumber{1300000}
% 正査
\advisor{正査教員 }
% 副査 がいる場合はコメントウ 名前を書く
% 副査 がない場合は、ここは削除しても可
```

```

%\coadvisor{副指導員 }
% 論議
\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}
% 和議 (2000 字程度)
\begin{jabstract} 日語 の概要を書く. (約 200 字)
\fake{ここに日語 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}
\tableofcontents % 目次
\listoffigures % 図表
\listoftables % 表表
%\doublespacing % ダブルスペース
\chapter{序論} % 章 (chapter) の例
ここに序論を書きます.
\section{背景} % 節 (section) の例
以下に背景 する環境 に関する概要を記述
\chapter{考察}
考察しました.
\section{評価 }
評価 をここに記述します.
\chapter{結論と今後の展開}
結論と今後の展望をここに記述します.
% 以降 (付録 )であることを示す
%\singlespacing % シングルスペース
\begin{appendix}
\chapter{アルゴリズム}
% 付録その 1(関数 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(関数 など)を必要があれば載せる
\chapter{ソース }
プログラム を 掲載します.
\section{ある }
何かを処理するあるプログラム\texttt{hoge.cpp}のプログラムを 示す.

```

```

\begin{verbatim}
int main( void ){ return 0; }
\end{verbatim}
\fake[40]{\thehoge} lines \par}
% 付録の終わり
\end{appendix}
\chapter*{謝辞}
謝辞を書く.
% 参蔵
\begin{thebibliography}{9}
\bibitem{ラベル} 著者 . 書籍 . 出版 , 年号.
\bibitem{MT1999} 未蔵 . 未来の未来. どこかの出版, 1999.
\end{thebibliography}
\end{document}

```

ファイルが大規模になる事が予想されているときは、記 のように`\include` 命令を使って、章毎に`\include` を分けると便利です。

```

%#!latex
\documentclass[dvipdfmx]{funthesis}[2004/11/10]
\usepackage{gsset}% 自作の設定ファイル gsset.sty を読み込む
%\includeonly{05con}% ある章だけダセめる
\begin{document}
\maketitle%          表紙
\input{00abst}%      概要
%\frontmatter %      前付け
\tableofcontents %   目次
\listoffigures %     図表
\listoftables %      表表
%\mainmatter %       本文
\include{01preface}% 序論
\include{02prevwork}% 関蔵
\include{03plan}%    提案する理論
\include{04exp}%     実験と評価
\include{05con}%     考察
\include{06postface}% 結論と今後の展開
\include{07thanx}%   謝辞
\include{08ref}%     参蔵
\appendix %          付録
\include{10algo}%    アルゴリズム
\end{document}

```

もし大学側からクラスが提供されていない場合は自前で作成することになります。しかも大抵の大学は Times 系のフォントを使ってフォントサイズは何々でという細かい指定をしてくるのが普通のようなので、観 な教員が作成してくれている場合もあります。とりあえず子供だましですが jsbook を用いた例を紹介します。jreport を使っても良いですが jsbook の方が個人的には良いと感じています。まずはご自分の大学の規定に合わせて jsbook に定義のいく

つかに変更を加えます。jsbook そのものに変更を加えるところ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。ファーム 先頭に

```
% Copying: Your Name
% E-mail: name@univ.ac.jp
% Date: 2004/02/20
\ProvidesPackage{mygs}[2004/03/31 First Family]
```

のようなファイル情報を書き込んでおくとよいでしょう。抵 の機関で Times 系のフォントを指定すると思いますので

```
\RequirePackage{txfonts}
```

の 1 行も必要でしょう。マクロパッケージの中で他のパッケージを必要とする場合は \Requirepacakge 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、定 の設定方法を紹介します。ページレイアウトで設定できる項目については図 B.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 で必要
```

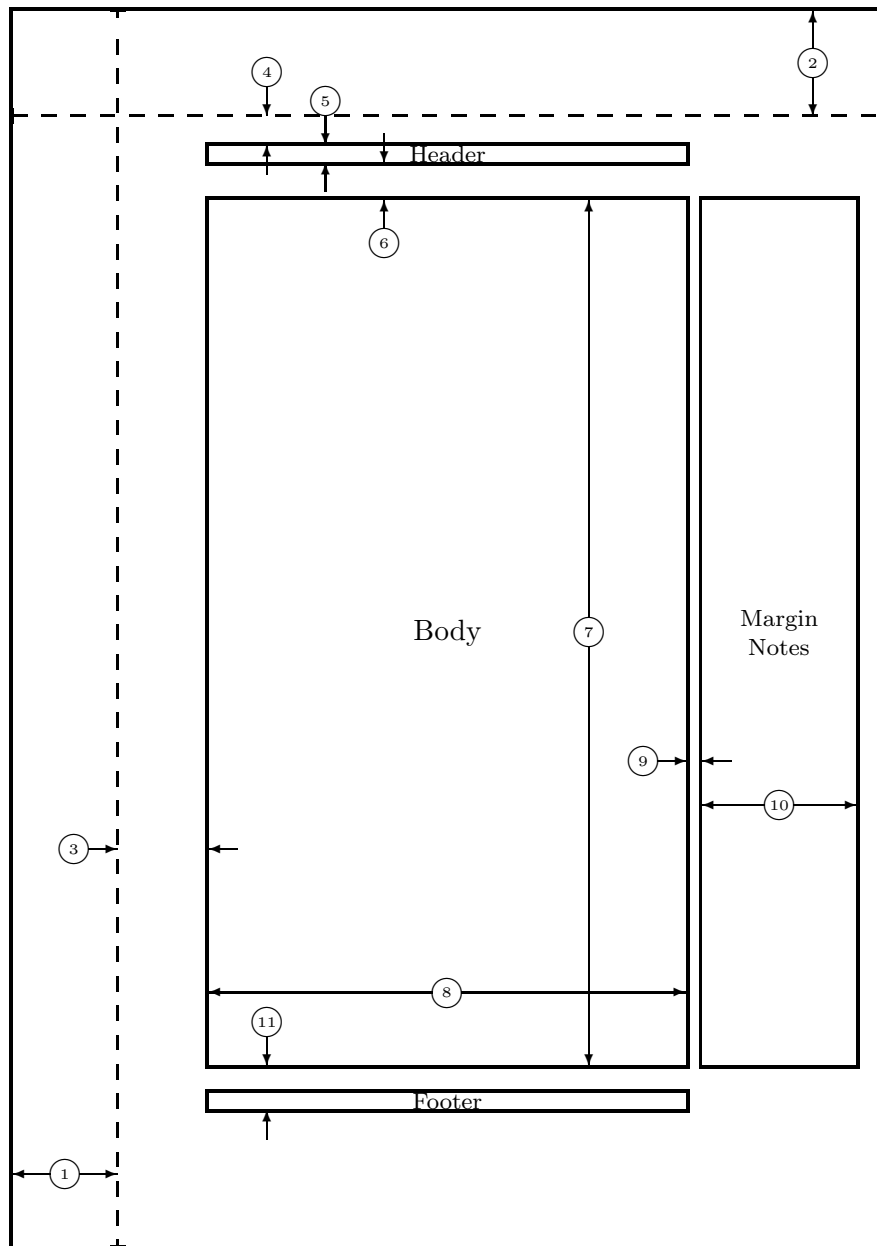
行数は 40 行と指定されている場合 \textheight に 40 行送り分 (40\baselineskip) を指定します。

```
\setlength\textheight{40\baselineskip}
```

この程度でも良いと思うのですが、

```
\setlength\hoffset{13\p@}
\setlength\voffset{0\p@}
\setlength\evensidemargin{0\p@}
\setlength\oddsidemargin{\evensidemargin}
\setlength\topmargin{0\p@}
\setlength\headheight{0\p@}
\setlength\headsep{0\p@}
\setlength\marginparwidth{0\p@}
\setlength\marginparpush{0\p@}
\setlength\marginparsep{0\p@}
```

のように設定しても良いでしょう。ここでの \p@ は単位 'pt' のことです。マクロの中ではこのような命令を使うと良いそうです。ここでは傍注やヘッダーを出力しないと仮定してほとんどの項目に '0pt' を代入しています。



- | | | | |
|----|------------------------------------|----|---|
| 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 B.1 版面のピア位 使用できる長さ

見出しのフォント 場合は和文はゴシック, 歐 は Times Bold としたい場合は jsbook の場合は

```
\renewcommand{\headfont}{\gtfamily\rmfamily\bfseries}
```

のようにしておけば良いでしょう. jsbook は標準では欧文がサンセリフ体になっています. jreport の場合は最初から欧文がゴシック体 に設定されています.

おまけに目次の深さを決めるコマンド tocdepth も

```
\setcounter{tocdepth}{2}
```

とすると \subsection まで出力されます.

jreport の場合は見出しの後の字下げが行われなことがあるので

```
\RequirePackage[indentfirst]
```

として indentfirst パッケージを読み込みます.

これらをまとめると自分のマクロパッケージ mygs.sty が出来 ます.

```
%% Copying : Thor Watanabe
%% E-mail : thor@tex.dante.jp
%% Date : 2004/02/20
\ProvidesPackage{mygs}[2004/02/20 First Family]
\RequirePackage{txfonts}% Times 系のフォントを使う
%\RequirePackage[indentfirst]% jreport は必要
\setlength\textwidth{40zw}%1 行 40 文字
\setlength\fullwidth{\textwidth}%jsbook では必要
\setlength\textheight{40\baselineskip}%1 ページ 40 行
\setlength\hoffset{13\p@}%\p@は 0pt のこと
\setlength\voffset{0\p@}
\setlength\evensidemargin{0\p@}
\setlength\oddsidemargin{\evensidemargin}
\setlength\topmargin{0\p@}
\setlength\headheight{0\p@}
\setlength\headsep{0\p@}
\setlength\marginparwidth{0\p@}
\setlength\marginparpush{0\p@}
\setlength\marginparsep{0\p@}
\setlength\footskip{2\baselineskip}% 必要に応じて
\def\ps@foot{% フッターに ‘-- ページ 番号 --’ としたいとき
\let\@mkboth\@gobbletwo
\let\@oddhead\@empty
\let\@evenhead\@empty
\def\@oddfoot{\normalfont\hfil-- \thepage\ --\hfil}%
\let\@evenfoot\@oddfoot
}
\pagestyle{plainfoot}%jsbook ならば
%\pagestyle{plain}%jreport ならば
\renewcommand{\headfont}{\normalfont\bfseries}
```

```
\setcounter{tocdepth}{2}
```

そのような作業が終わったら自分の論文の主となるマスタイルを 書き上げます。紙 は A4 で、フォントサイズは 11pt, 左右起こしはせずに片面印刷というのが一般的だと思いますから

```
\documentclass[a4j,11pt,openany,oneside]{jsbook}
```

のようにします。そして先 した mygs.sty を

```
\usepackage{mygs}
```

として読み込みます。

この程度でも良いのですが、紙 もまた細かい指定をされる場合があります。1 から \maketitle を作っても良いのですが、一刻も早く論文を仕上げなければならないときに、命令を定義しては間に合わないかも知れません。そのようなときは断腸の思いで \titlepage 環境を借用して表紙を作ることもできます。例として \maketitle 命令の変更例を紹介し

ます。

```
\renewcommand{\maketitle}{%
\begin{titlepage}
\let\footnotesize\small
\let\footnoterule\relax
\let\footnote\thanks
\null\vskip2em% ペジ 上部の空白
\begin{center}\thispagestyle{empty}%
{\LARGE\headfont ここに表題を書きます}\par\vskip1.5em
{\Large\normalfont 未  }\par\vskip2em
{\small 未  \quad 学籍  }\par\vskip1em
{\small 指  \quad 北  }\par\vskip2em
{提出 2004/02/30}\par\vskip1em
{\Large\headfont English Title}\par\vskip1em
{\large\rmfamily Your Name}\par\vskip1em
\end{center}%
\vfill\null
\end{titlepage}}
```

\vskip とは垂直 に空気を挿入する命令です。

以上は例ですので先方に規定された通りの 適 してください。

GNU Free Documentation License

Version 1.2, November 2002
Copyright © 2000, 2001, 2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall sub-

ject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this

definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and dis-

tribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to

address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.



参 照

- [1] 江口 肇 . *Ghostscript Another Manual*. ソフトバンク, 1997
- [2] Michel Goossens and Sebastian Rahtz. *L^AT_EX Web コンパニオン——T_EX と HTML/XML の統合*. アスキ , 2001. 鷺 野
- [3] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX コンパニオン*. アスキ , 1998. 021.49/Go
- [4] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *L^AT_EX グラフックスコンパニオン*. アスキ , 2000. 鷺 野 , 007.63/Go
- [5] George Gratzer. *Math into L^AT_EX*. Birkhauser, 2000. 021.49/Gr
- [6] 日本エディタースクール編. *校正記号の使い方——タテ組・ヨコ組*. 日本エディタースクール , 1999
- [7] 日本エディタースクール編. *文字の組方ルールブック〈ヨコ組編〉*. 日本エディタースクール , 2001
- [8] 藤田 眞作. *X_YL^AT_EX: typesetting chemical structural formulas*. 星雲社, 1997. 430.7/Fu
- [9] 藤田 肇 . *L^AT_EX 2_ε コマンドブック*. ソフトバンク, 2003
- [10] Cho Jin-Hwan. *DVIPDFMx, an eXtension of DVIPDFM*, 2003.
▶ <http://project.ktug.or.kr/dvipdfmx/>
- [11] 木 雁 . *理 科 の 作 業* . 中 新 書 624. 中 文 学 , 1981. 080//Ch//624
- [12] Donald Knuth. *改 訂 T_EX ブック*. アスキ , 1992. 斎 藤 , 鷺 野
- [13] Leslie Lamport. *文書処理システム L^AT_EX 2_ε*. ピアソン・エデュケーション, 1999. 阿瀬はる美訳, 021.4//La
- [14] 松 垂 . *日 語 B_BT_EX: jB_BT_EX*, 1991. jB_BT_EX と共に配布される文書
- [15] 小林道正, 小林研. *L^AT_EX で数学を——L^AT_EX 2_ε + A_MS-L^AT_EX 入門*. 朝倉書店, 1997. 410.7//Ko
- [16] 中田英雄, 金城悟編. *大学生のための研究論文のまとめ方——データ収集からプレゼンテーションまで* . 文 芸 , 1998. 002.7//Na
- [17] Tobias Oetiker. *L^AT_EX 2_ε への道——94分 L^AT_EX 2_ε 入門* , 2000. 野 間 ,
▶ <http://www.ring.gr.jp/pub/text/CTAN/info/lshort/japanese/>
- [18] Tobias Oetiker. *The Not So Short Introduction to L^AT_EX 2_ε*, 2003.
▶ <http://www.ring.gr.jp/pub/text/CTAN/info/lshort/english/>
- [19] 小 藤 . *大 学 の た め の トポロジ* . 講 談 社 , 2002. 080//Ko//1603
- [20] 大 塚 . *分 かり や す い 日 語 の 書 き 方*. 講 談 社 , 2004. 080//Ko//1644

- [21] 奥 隼 . [改訂 3 版] $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 美 鑑 . 技 薈 , 2004. 021.49/Ok
- [22] 乙部 巖己, 江口庄英. *pL^AT_EX 2_ε for Windows Another Manual Vol.2 Extended Kit 1997*. ソフトバンク, 1997. 021.49/Ot
- [23] 乙部 巖己, 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. 021.49/Ot
- [24] Oren Patashnik. $\text{B}\text{I}\text{B}\text{T}_{\text{E}}\text{X}\text{ing}$: $\text{B}\text{I}\text{B}\text{T}_{\text{E}}\text{X}$ の使い方, 1991. 松井正一訳, $\text{j}\text{B}\text{I}\text{B}\text{T}_{\text{E}}\text{X}$ と共に配布される文書
- [25] Keith Reckdahl. *Using Imported Graphics in L^AT_EX 2_ε*, 1997.
▶ <http://www.ring.gr.jp/archives/text/CTAN/info/epslatex.pdf>
- [26] 嶋田 隆 . $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 数 躰 境 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ を使いこなす. シイム ・ シイ出版部, 2001. 410//Sh
- [27] 白 嗣 , 伊 藤 , 井上 槌 . *Linux 論 叢* . オム 社, 1999. 816.5/Us
- [28] Mark A. Wicks. *Dvipdfm User's Manual*, 1999.
▶ <http://gaspra.kettering.edu/dvipdfm/>
- [29] Thomas Williams and Colin Kelley. *Gnuplot, An Interactive Plotting Program*, 2003.
▶ <http://www.gnuplot.info/>
- [30] Timothy Van Zandt. *PSTricks: PostScript macros for Generic T_EX*, 1993.
▶ <http://www.tug.org/applications/PSTricks/>

索引

- 数字/記号
- \□ 33
 - 数式 の—— 81
 - 文中の—— 132
 - 28, 68
 - \! 81, 132
 - \" (ii) 28
 - \# 8
 - # 8, 68
 - \\$ 8
 - \$ 8, 10, 68, 77
 - コンソール ——— ii
 - \$\$ 78
 - \% 8
 - % 8, 29, 68
 - \& 8
 - & 8, 68
 - array 環境の—— 87
 - eqnarray*環境の—— 79
 - tabular 環境の—— .. 105
 - \' (é) 28
 - ' 30
 - \(..... 77
 - (..... 85
 - \) 77
 -) 85
 - * 12
 - \, 81, 132
 - , (読点) 26
 - 31
 - ハオとしての —— 31
 - マオとしての —— 31
 - 31
 - 31
 - \. (à) 28
 - . (短 3点リダ) 94
 - ... (桁 3点リダ) 94
 - / 86, 100
 - 区切り記号の—— 86
 - 分数の—— 89
 - \: 81, 132
 - \; 81, 132
 - < 8
 - \= (ã) 28
 - > 8
 - @ 67
 - \@ 33
 - \[..... 78
 - [..... 85
 - \ 8, 64, 68
 - \] 78
 -] 85
 - \^ (ô) 28
 - ^ 8, 68, 82
 - _ 8
 - _ 8, 68, 82
 - \` (à) 28
 - ` 30
 - \{ 8, 85
 - { 8, 68
 - \} 8, 85
 - } 8, 68
 - \| 85
 - | 8, 85
 - \~ (ñ) 28
 - ~ 8, 34, 68, 132
 - 10pt 25, 44, 45
 - 11pt 44, 45
 - 12pt 44, 45
 - 12Q 45
 - 14pt 45
 - 14Q 45
 - 17pt 45
 - 20pt 45
 - 21pt 45
 - 25pt 45
 - 2項籟子 83
 - 2段組 127, 139
 - の段間の罫線 128
 - のときの段間 128
 - 30pt 45
 - 36pt 45
 - 3点リダ 94
 - 下付き—— 94
 - 中点 94
 - 43pt 45
 - 9pt 45
- A**
- a4j 44, 45
 - a4paper 44, 45
 - a4var 45
 - a5j 44, 45
 - a5paper 44, 45
 - a6paper 45
 - \AA (Å) 28
 - \aa (å) 28
 - abbrv 53
 - abstract 139
 - abstract 21, 23
 - abstract 環境 23
 - Acrobat Reader 61
 - \acute (á) 93
 - \addcontentsline 23
 - address (BIBTEX) 54
 - \addtocounter 73
 - \addvspace 133
 - Adobe 60
 - Adobe Acrobat 119, 121
 - Adobe Reader 60, 61
 - Adobe-Japan1-6 138
 - \AE (Æ) 28
 - \ae (æ) 28
 - \AffixLabels 122
 - \afterpage 46
 - afterpage 46
 - Alan Jeffery iii
 - Aleph 138
 - \aleph (ℵ) 94
 - \Alph 73
 - \alph 73
 - \alpha (α) 90
 - alpha 53
 - \amalg (II) 92
 - American Mathematical Society 45
 - amount of substance 26
 - ampere 26
 - amsbsy 99
 - amsfonts 94
 - \AMS-ITEX 45
 - amsmath 45
 - amssymb 80, 94, 120
 - \AMS-TeX 45
 - amsthm 95

- `\angle` (\angle) 94
`\annotate` (BIBTEX) 54
`\appendix` 134
`\appendixname` 128
`appendix` 環境 134
`\approx` (\approx) 92
`\arabic` 73
`\arccos` (\arccos) 83
`\arcsin` (\arcsin) 83
`\arctan` (\arctan) 83
`\arg` (\arg) 83
`array` 46
`array` 環境 .. 46, 86, 105, 109
`\Arrowvert` (\Uparrow) 85
`\arrowvert` (\Downarrow) 85
`article` 21, 43, 139
`article` (齣の種類) ... 53
`ASCII` 138
`\ast` ($*$) 92
`\asymp` (\asymp) 92
`\AtBeginDvi` 62
`\atop` 101
`ATSUI` 137
`\author` 19, 20
`author` (BIBTEX) .. 51, 54, 56
`author` 20
`.aux` (拡張) 16
- ## B
- `\b` (\bar{a}) 28
`\B` 122
`b4j` 44, 45
`b4paper` 44, 45
`b5j` 44, 45
`b5paper` 44, 45
`b5var` 45
`babel` 45
`\backslash` (\backslash) 85, 94
`balance` 129
`\bar` (\bar{a}) 93
`\baselineskip` 147
`.bb` (拡張) 16
`.bbl` (拡張) 16
`\begin` 14
`\beta` (β) 90
`\bfseries` 41
`.bib` (拡張) 16
`\bibitem` 47, 48, 49
`\bibliography` 51, 52
`\bibliographystyle` .. 51, 52
`\bibname` 128
`BIBTEX` 47
`address` 54
`annotate` 54
`author` 51, 54, 56
`booktitle` 54
`chapter` 54
`crossref` 54
`edition` 54
`editor` 53, 54
`howpublished` 54, 56
`journal` 54, 55
`key` 54
`month` 54, 56
`note` 51, 54
`number` 54, 55
`organization` 54
`pages` 54, 55
`publisher` 51, 54
`school` 54
`series` 54
`title` 51, 54
`volume` 54
`year` 51, 54
`yomi` 51, 55
`\Big` 86
`\big` 86
`\bigcap` (\cap) 92
`\bigcirc` (\bigcirc) 92
`\bigcup` (\cup) 92
`\bigl` 86
`\bigm` 86
`\bigodot` (\odot) 92
`\bigoplus` (\oplus) 92
`\bigotimes` (\otimes) 92
`\bigr` 86
`\bigskip` 133
`\bigsqcup` (\sqcup) 92
`\bigtriangledown` (∇) .. 92
`\bigtriangleup` (\triangle) 92
`\biguplus` (\uplus) 92
`\bigvee` (\vee) 92
`\bigwedge` (\wedge) 92
`.blg` (拡張) 16
`\bm` 99
`bm` 46, 99
`\bmod` 83
`BMP` 113
`.bmp` (拡張) 16
`\boldmath` 99
`\boldsymbol` 99
`book` (齣の種類) ... 53, 54
`booklet` (齣の種類) ... 53
`booktabs` 107, 108
`booktitle` (BIBTEX) 54
`\bordermatrix` 88
`\bot` (\perp) 94
`\bottomrule` 108
`\bowtie` (\bowtie) 92
`\Box` (\square) 94
`\brace` 101
`\bracket` 101
`\breve` (\acute{a}) 93
`.bst` (拡張) 16
`\bullet` (\bullet) 92
- ## C
- `\c` (\c) 28
`Calc` 110
`calc` 46
`Calc2LATEX` 110
`candela` 26
`\cap` (\cap) 92
`\caption` 73, 104
`\cases` 101
`cases` 環境 101
`\catcode` 67
`cd` 8
`\cdot` (\cdot) 92
`\cdots` (\cdots) 94
`\centering` 38
`centering` 38
`center` 環境 38
`\chapter` 21, 73, 144
`chapter` (BIBTEX) 54
`chapter` (カウタ) 72
`chapter` 19, 21
`\chapter*` 23
`character` 19, 39
`\check` (\check{a}) 93
`\chi` (χ) 90
`Cho Jin-Hwan` iii, 61
`\choose` 101
`Chris Rowley` iii
`\circ` (\circ) 92
`\cite` 46, 47, 48, 51
`cite` 57
`\citeform` 58
`\citeleft` 58
`\citimid` 58
`\citepunct` 58
`\citeright` 58
`class` 43
`\cleardoublepage` 127
`\clearpage` 46, 127
`\cline` 88, 106
`.clo` (拡張) 15
`.cls` (拡張) 15
`\clubsuit` (\clubsuit) 94
`\cmidrule` 108
`Colin Kelley` 123
`color` 45
`\columnsep` 127, 129
`\columnseprule` 127, 129
`\columnwidth` 129
`comment` 29
`comment out` 29

- comment 環境 29
Computer Modern 137
conference 56
config.pdf 62
config.ps 62
\cong (\cong) 92
contents 19, 21, 22
\contentsname 128
\coprod (\coprod) 92
copy 8
\cos (cos) 83
\cosh (cosh) 83
\cot (cot) 83
\coth (coth) 83
cp 8
\cr 88
\crrc 98
CreateBB 16
crossref (BibTeX) 54
\csc (csc) 83
\cup (\cup) 92
- ## D
- \d (\d) 28
\dag (\dagger) 28
\dagger (\dagger) 92
\dashv (\dashv) 92
\date 19, 20
date 20
David Carlisle iii, 76, 88, 109
dcolumn 46, 109
\ddag (\ddagger) 28
\ddagger (\ddagger) 92
\ddot ($\ddot{\cdot}$) 93
\ddots (\ddots) 94
\deg (deg) 83
del 8
delarray 46, 88
delimiter array 88
\Delta (Δ) 91
\delta (δ) 90
description 環境 35
\det (det) 83
\DH (\mathbb{D}) 28
\dh (δ) 28
\Diamond (\diamond) 94
\diamond (\diamond) 92
\diamondsuit (\diamondsuit) 94
\dim (dim) 83
dir 8
displaymath 環境 78
\displaystyle 90
\div (\div) 92
\DJ (\mathbb{D}) 28
\dj (\mathbb{d}) 28
document 19
\documentclass 14, 16
document 環境 14
Donald Arseneau 57, 136
Donald Knuth iii, 3, 137
\dot ($\dot{\cdot}$) 93
\doteq (\doteq) 92
\doublespacing 25
\Downarrow (\Downarrow) 85, 93
\downarrow (\downarrow) 85, 93
draft 44, 122
.dtx (拡張) 15
DVI 5, 7, 60
—の再素 12
.dvi (拡張) 16
Dviout 12
Dvipdfm 16, 61
dvipdfm 112
Dvipdfmx 61
dvipdfmx 112, 113, 118
dvipdfmx.def 112
dvips 112, 113, 118
dvipsk 62
- ## E
- \E 122
e-mail 136
EasyPackage 4
EasyTeX 18
ebb 16, 113
edition (BibTeX) 54
editor (BibTeX) 53, 54
eepic 111, 120
electric current 26
\ell (ℓ) 94
em-dash 31
Emacs 5, 6, 18
\emph 27, 31
emphasis 27
\emptyset (\emptyset) 94
en-dash 31
end 14
\endSetLabels 122
english 45, 138
\enskip 132
\enspace 132
enumerate 46
enumerate 環境 35, 46
enumi (カウタ) 72
enumii (カウタ) 72
enumiii (カウタ) 72
enumiv (カウタ) 72
EPDF 111, 138
epic 111, 120
.eps (拡張) 16
\epsilon (ϵ) 90
epstopdf 117
eqnarray*環境 79
eqnarray 環境 73, 79
\equation 73
equation (カウタ) 72
equation 環境 79
\equiv (\equiv) 92
\eta (η) 90
 ε -TeX 137
\evensidemargin 125, 147
\example 65
Excel 110, 111
exceltex 111
Exel2tabular 110
\exists (\exists) 94
\exp (exp) 83
- ## F
- \fbox 130
\fbboxrule 130
\fbboxsep 130
.fd (拡張) 15
Fedora Core 12, 62
figure (カウタ) 72
figure 104
figure*環境 129
\figurename 128
figure 環境 104
file 113
final 44, 122
\flat (\flat) 94
fleqn 44, 78
float 104
flushing 38
flushleft 38
flushleft 環境 38
flushright 38
flushright 環境 38
\fnsymbol 73
fontenc 28
\footnote 27
footnote (カウタ) 72
\footnotemark 107
\footnotesize 40
\footnotetext 107
\footskip 125
\forall (\forall) 94
Foxit Reader 60
Foxit Software Company 60
\frac 84, 98, 100
\framebox 130
Frank Mittelbach .. iii, 94, 95
\frontmatter 23
\frown (\frown) 92
ftnright 46
FUNNIST i
funthesis 144

G

\backslash Gamma (Γ) 91
 \backslash gamma (γ) 90
 \backslash gcd (gcd) 83
 \backslash ge (\geq) 92
 Geoffrey Tobin 25
 geometry 127
 \backslash gg (\gg) 92
 Ghostscript 60
 giga 26
 GNU
 —Free Documentation License 2
 —フリー文書離約書 2
 gnu-head.pdf 115
 Gnuplot 18, 123
 GPL 120
 Grapher 120
 graphicx 17, 45
 \backslash grave (\grave{a}) 93
 \backslash GridLineWidth 122
 \backslash gtfamily 42
 \backslash guillemotleft (\llcorner) 28
 \backslash guillemotright (\ggcorner) 28
 \backslash guilsinglleft (\langle) 28
 \backslash guilsinglright (\rangle) 28

H

\backslash H (\hat{a}) 28
 Han The Thanh 137
 Hans-Peter Doerr 111
 Harald harders 82
 \backslash hat (\hat{a}) 93
 \backslash hbar (\hbar) 94
 \backslash headheight 125, 147
 headline 19
 \backslash headsep 125, 147
 \backslash heartsuit (\heartsuit) 94
 help 8
 \backslash hfill 121
 hline 46
 \backslash hline 88, 106
 \backslash hoffset 125, 147
 \backslash hom (hom) 83
 \backslash hookleftarrow (\hookleftarrow) 93
 \backslash hookrightarrow (\hookrightarrow) 93
 \backslash howpublished (BIBTEX) 54, 56
 \backslash hphantom 100
 \backslash hspace 127
 \backslash hspace* 127, 133
 \backslash hss 98
 HTML 4, 60, 63
 \backslash Huge 40
 \backslash huge 40

Hyper Link 60
 HyperTeX 61

I

\backslash i (i) 28
 identify 113
 Illustrator 119, 120
 \backslash Im (\Im) 94
 \backslash imath (i) 94
 \backslash in (\in) 92
 inbook (蔵の種類) 53
 \backslash include 134, 146
 \backslash includegraphics .. 113, 114, 117, 118
 \backslash includeonly 134, 146
 incollection (蔵の種類) 53
 \backslash indent 24
 indentation 19, 23
 indentfirst 24, 46, 149
 \backslash inf (inf) 83
 info 9
 \backslash infty (∞) 94
 Ingo H. de Boer 18
 inproceedings (文献の種類) 53, 56
 \backslash input 120, 134
 .ins (拡張) 15
 \backslash int (\int) 92
 \backslash iota (i) 90
 \backslash item 35
 itemization 35
 itemize 環境 35
 \backslash itshape 41

J

\backslash j (j) 28
 jabbrv 53
 jalpha 53
 jarticle 17, 21, 43, 46, 138, 139
 jBIBTEX 47, 50
 jbook 44, 138
 jclasses 44
 Jerry Jeichter 109
 JIS X 0208 138
 JIS 第2水準 138
 \backslash jmath (j) 94
 Johannes Braams iii
 John Eaton 124
 \backslash Join (\bowtie) 94
 journal (BIBTEX) 54, 55
 JPEG 113
 .jpg (拡張) 16
 jplain 53
 jplain.bst 52
 jreport 44, 46, 138, 144
 jsarticle 44, 138, 139

jsbook 44, 138, 144, 146
 jsclasses .. 4, 44, 45, 133, 138
 jspfd 44
 junsrt 53

K

\backslash k (o) 28
 \backslash kappa (κ) 90
 Karl Berry 62
 kelvin 26
 \backslash ker (ker) 83
 kerning 33
 key (BIBTEX) 54
 Keynotes 120
 kilo 26
 kilogram 26
 Kpathsearch 62
 Kristoffer Rose 124

L

\backslash L (L) 28
 \backslash l (l) 28
 \backslash L 122
 \backslash label ... 46, 71, 76, 104, 105
 Label ‘key’ multiply defined 76
 Label(s) may have changed. Return to get cross-ferences right. 76
 labelfig 122
 Lambda 138
 \backslash Lambda (Λ) 91
 \backslash lambda (λ) 90
 Lamed 138
 landscape 44
 \backslash langle (\langle) 85
 \backslash LARGE 40
 \backslash Large 40
 \backslash large 40
 \backslash LATEX 3
 —原稿 7
 —纏 7
 —での計算 46
 —と \mathcal{P} の 違い .. 24
 —における単位 25
 —に関わる \mathcal{R} イ 15
 —の動かし方 5
 —のエラ 9
 —の基本 5
 —の最観 の規則 1
 —の作者 iii
 —の実行 5
 —の周報 i
 —の専門 iv
 —の中途ファイル 7

- の導入 i
 - の日暮 4
 - の入替 iv
 - の入換 8
 - の発音 i
 - のログファイル 7
 - \LaTeX 34
 - latex 6
 - latexmk 52
 - latexsym 94
 - Laurent Siebenmann 122
 - \layout 125
 - layout 46, 125
 - \lbrace ({) 85
 - \lceil (⌈) 85
 - \ldots (...) 94
 - \le (≤) 92
 - leading 34
 - \leadsto (↷) 94
 - \left 84, 85
 - \left(..... 88
 - \Leftarrow (⇐) 93
 - \leftarrow (←) 93
 - \leftharpoondown (⇽) 93
 - \leftharpoonup (⇵) 93
 - \leftidx 82
 - leftidx 82
 - \Leftrightarrow (⇔) 93
 - \leftrightarrow (↔) 93
 - length 26
 - leqno 44
 - Leslie Lamport iii, 3, 4
 - less 9
 - letter 19
 - space 34
 - letterpaper 44
 - letter サネ 62
 - \lfloor (⌊) 85
 - \lggroup (⏟) 85
 - \lhd (⋈) 94
 - ligature 33
 - \lim (lim) 83
 - \liminf (lim inf) 83
 - \limits 84
 - \limsup (lim sup) 83
 - line break 32
 - List of Figures 22
 - List of Tables 22
 - \listfigurename 128
 - \listoffigures 22
 - \listoftables 22
 - \listtablename 128
 - \ll (≪) 92
 - .lof (拡張) 16
 - .log (log) 83
 - .log (拡張) 16
 - \Longleftarrow (⇐) 93
 - \longleftarrow (←) 93
 - \Longleftrightarrow (⇔) 93
 - 93
 - \longmapsto (↦) 93
 - \Longrightarrow (⇒) 93
 - \longrightrightarrow (⇨) 93
 - longtable 46
 - .lot (拡張) 16
 - ls 8
 - ltboxes.dtx 129
 - \ltrans 82
 - luminous intensity 26
- ## M
- Mac OS X 137
 - でのプレビュー 60
 - のATSUI 137
 - の執筆環境 18
 - への導入 4
 - MacOS X WorkShop 4
 - MacWiki 4
 - Make 18, 52
 - \makeatletter 67
 - \makeatother 67
 - \makebox 129
 - Makefile 18
 - \maketitle 20, 23, 150
 - man 8
 - manual (敵 の種類) 53
 - \mapsto (↦) 93
 - \marginpar 27
 - \marginparpush 127, 147
 - \marginparsep 127, 147
 - \marginparwidth 127, 147
 - Mark Wicks iii, 61
 - mass 26
 - masterthesis (敵 の種類) 53
 - \mathbb 81
 - \mathbf 80, 99
 - \mathcal 80
 - Mathematica 120
 - \mathfrak 81
 - \mathit 2, 80
 - \mathnormal 80
 - \mathrm 80, 91
 - \mathsf 80
 - \mathstrut 99
 - \mathtt 80
 - math 環境 78
 - MATLAB 120, 121
 - \matrix 88
 - matrix 環境 88
 - \max (max) 83
 - \mbox 130
 - \mcfamily 42
 - \mdseries 41
 - \medskip 133
 - mega 26
 - METAFONT 120
 - METAPOST 16, 120
 - meter 26
 - \mho (Ω) 94
 - Michael Downes iii
 - micro 26
 - Microsoft Office 3
 - \mid (|) 92
 - \midrule 108
 - milli 26
 - \min (min) 83
 - minipage 環境 121, 131
 - misc (敵 の種類) 53
 - Misplaced alignment tab character & 10
 - Missing \$ inserted 10
 - mkdir 8
 - \models (⊨) 92
 - mole 26
 - month (BibTeX) 54, 56
 - morisawa 138
 - move 8
 - \mp (∓) 92
 - .mp (拡張) 16
 - mpfootnote (カウタ) 72
 - \mu (μ) 90
 - multicol 46
 - \multicolumn 88, 106, 108
 - \multirow 109
 - multirow 109
 - mv 8
 - Mxdvi 60
- ## N
- \nabla (∇) 94
 - nano 26
 - \natural (♮) 94
 - \nearrow (↗) 93
 - \neg (¬) 94
 - \negthinspace 132
 - Nelson Beebe iii
 - \neq (≠) 92
 - nest 35
 - nested sections 21
 - \newcolumntype 109
 - \newcommand 65
 - \newcommand* 136
 - \newcounter 73
 - \newenvironment 66
 - \newlabel 76
 - \newline 32
 - \newpage 127

- `\newtheorem` 94, 95
`\NG` (\mathbb{N}) 28
`\ng` (\mathbb{N}) 28
`\ni` (\ni) 92
`\nocite` 52
`\noindent` 24
`\nolimits` 84
`\nonumber` 80
`\normalsize` 40
`nosort` 57
`nospace` 57
`\not` 92, 98
`note` (`\BIBTeX`) 51, 54
`note` 19, 27
`\notin` (\notin) 92
`notitlepage` 44
`\nu` (ν) 90
`number` (`\BIBTeX`) 54, 55
`\narrow` (\searrow) 93
- O**
- `\O` (\mathbb{O}) 28
`\o` (\mathbb{O}) 28
`obsolete` 18
`Octave` 120, 124
`\oddsidemargin` 125, 147
`\odot` (\odot) 92
`\OE` (\mathbb{E}) 28
`\oe` (\mathbb{E}) 28
`\oint` (\oint) 92
`okumacro` 31, 138
`okuverb` 138
`Omega` 138
`\Omega` (Ω) 91
`\omega` (ω) 90
`\ominus` (\ominus) 92
`OmniGraffle` 120
`\onecolabstract` 139
`\onecolumn` 129
`onecolumn` 44
`\onehalfspacing` 25
`oneside` 44, 125
`\oalign` 98
`openany` 44
`OpenOffice.org` 3
`openright` 44
`\oplus` (\oplus) 92
`Oren Patashnik` iii, 47
`organization` (`\BIBTeX`) .. 54
`\oslash` (\oslash) 92
`OTF` 138
`\otimes` (\otimes) 92
`\overbrace` 93
`Overfull` 11
`\overleftarrow` 93
`\overline` 93
- `\overrightarrow` 93
- P**
- `\P` (\mathbb{P}) 28
`package` 43
`page` (`\Cout`) 72
`\pageref` 71, 76
`Pages` 120
`pages` (`\BIBTeX`) 54, 55
`Palatino` 138
`papersize` 45
`\par` 24, 32, 34
`\paragraph` 21
`paragraph` (`\Cout`) 72
`paragraph` 19, 22
`—skip` 34
`\parallel` (\parallel) 92
`\parbox` 130
`\parindent` 24
`\part` 21
`part` (`\Cout`) 72
`part` 19, 21
`\partial` (∂) 94
`PATH` 117
`\path` 136
`PDF` 60, 113
`—齧の張り込み` .. 113
`—と Mac OS X` 120
`—のフォント` .. 59, 61
`—のプレビュー` 60
`—の編集` 121
`—ブックマーク` 61
`PDF ϵ -L \TeX` 137
`PDF ϵ -T \TeX` 137
`PDFL \TeX` 60, 137
`PDFT \TeX` 137
`pdvips` 62
`Pehong Chen` iii
`Perl` 111
`\perp` (\perp) 92
`\phantom` 100
`phdthesis` (`\Cout` の種類) . 53
`\Phi` (Φ) 91
`\phi` (ϕ) 90
`Photoshop` 120
`\Pi` (\mathbb{P}) 91
`\pi` (π) 90
`pico` 26
`pict2e` 111
`PicT \TeX` 120
`picture` 環境 104, 111
`Piet Van Oostrum` 109
`plain` 53
`pL \TeX` 4
`platex` 6
`\pm` (\pm) 92
- `\pmatrix` 88
`pmatrix` 環境 88
`\pmod` 83
`PNG` 113
`.png` (拡張子) 16
`\postchaptername` 128
`\postpartname` 128
`PostScript` 60
`\postsectionname` 128
`\pounds` (\pounds) 28
`PPM` 112
`\Pr` (\Pr) 83
`\prec` (\prec) 92
`\preceq` (\preceq) 92
`\prechaptername` 128
`\prepartname` 128
`\presectionname` 128
`\prime` (\prime) 94
`PrimoPDF` 119
`proc` 43
`proceeding` 56
`\prod` (\prod) 92
`\propto` (\propto) 92
`\ProvidesPackage` 147
`ps2jpdf` 117
`ps2pdf` 117
`\Psi` (Ψ) 91
`\psi` (ψ) 90
`PStricks` 120
`pT \TeX` 4
`publisher` (`\BIBTeX`) .. 51, 54
`punctuation` 19, 26
`pxdvi` 12
`pxfonts` 99, 138
- Q**
- `QPL` 120
`\qqquad` 81, 132
`\quad` 81, 132
`quotation` 環境 30
`\quotedblbase` („) 28
`\quotesinglbase` (‘) 28
`quote` 環境 30
- R**
- `R` 120
`\r` (δ) 28
`\R` 122
`\raggedleft` 38
`\raggedright` 38
`Rainer Schöpf` iii
`\rangle` (\rangle) 85
`Raymond Séroul` 122
`\rbrace` ($\}$) 85
`\rceil` (\lceil) 85
`\Re` (\Re) 94
`Red Hat` 12, 62

- `\ref` 46, 71, 76, 105
 Reference ‘key’ on page n
 undefined 76
 references 47
`\reflectbox` 116
`\refname` 128
`\refstepcounter` 73
`\renewcommand` 65
`\renewenvironment` 66
 report 43
 report 45, 138
`\RequirePackage` 147
`\resizebox` 116
`\rfloor` (J) 85
`\rgroup` (J) 85
`\rhd` (▷) 94
`\rho` (ρ) 90
 Richard Koch 18
`\right` 84, 85
`\right)` 88
`\Rrightarrow` (\Rightarrow) 93
`\rightarrow` (\rightarrow) 93
`\rightharpoonowdown` (\rightharpoonowdown) 93
`\rightharpoonoup` (\rightharpoonoup) 93
`\rightleftharpoons` (\rightleftharpoons) 93
 rm 8
`\rmfamily` 41
`\Roman` 73
`\roman` 73
 root 4
 Ross Moore 124
`\rotatebox` 116
 Rplots.pdf 120
`\rule` 131
- S**
- `\S` (§) 28
`\sb` 82
`\scalebox` 116
 school (BIBTEX) 54
 SciLab 124
`\scriptscriptstyle` 90
`\scriptsize` 40
`\scriptstyle` 90
`\scshape` 41
`\searrow` (\searrow) 93
 Sebastian Rahtz iii
`\sec` (sec) 83
 secnumdepth (カウタ) 22
 second 26
`\section` 21, 73, 144
 section (カウタ) 72
 section 19, 21
 sectioning 21
 sentence 19
 —space 34
 serial number 21
 series (BIBTEX) 54
`\setcounter` 73
`\SetLabels` 122
`\setminus` (\setminus) 92
 setspace 25
`\sffamily` 41
`\sharp` (#) 94
`\ShowGrid` 122
 showkeys 46, 76
 SI 26
`\Sigma` (Σ) 91
`\sigma` (σ) 90
`\sim` (\sim) 92
`\simeq` (\simeq) 92
 Simon Fear 108
`\sin` (sin) 83
`\singlespacing` 25
`\sinh` (sinh) 83
 slides 43
`\slshape` 41
`\small` 40
`\smallskip` 133
`\smash` 100
`\smile` (\smile) 92
`\sp` 82
 space 57
 space 34
 letter— 34
 sentence— 34
 word— 34
 spacing 環境 25
`\spadesuit` (♠) 94
`\sqcap` (\sqcap) 92
`\sqcup` (\sqcup) 92
`\sqrt` 84
`\sqsubset` (\sqsubset) 94
`\sqsubseteq` (\sqsubseteq) 92
`\sqsupset` (\sqsupset) 94
`\sqsupseteq` (\sqsupseteq) 92
`\SS` (SS) 28
`\ss` (ss) 28
`\stackrel` 98
`\star` (*) 92
`\stepcounter` 73
 Steven Douglas Cochran 121
`\strut` 122
 .sty (拡張) 15
 subfigure 121
`\subparagraph` 21
 subparagraph (カウタ) 72
 subparagraph 22
`\subsection` 21, 144
 subsection (カウタ) 72
 subsection 21
`\subset` (\subset) 92
- T**
- `\t` (oo) 28
`\T` 122
 T1 28
 tabbing 環境 105
`\table` 73
 table (カウタ) 72
 table 104
 table*環境 129
`\tablename` 128
`\tableofcontents` 22, 128
 table 環境 104
 tabular*環境 46
 tabularx 46
 tabular 環境 46, 104, 105,
 109
`\tan` (tan) 83
`\tanh` (tanh) 83
`\tau` (τ) 90
 tera 26
 TeX 3
 —の読み方 3
`\TeX` 34
 .tex (拡張) 16
 TeXShop 18
`\text` 80, 81
`\textasciicircum` 8
`\textasciitilde` 8
`\textbackslash` 8
`\textbar` 8
`\textbf` 41
 textcomp 26
`\textgreater` 8
`\textgt` 42
`\textheight` 127, 147
`\textit` 41
`\textless` 8
`\textmc` 42
`\textmd` 41
`\textmu` 26
`\textquotedbl` (") 28

- `\textrm` 41
`\textsc` 41
`\textsf` 41
`\textsl` 41
`\textstyle` 90
`\texttt` 41
`\textwidth` 127, 129, 147
 Tgif 18, 120
`\TH` (b) 28
`\th` (b) 28
 thebibliography 環境 16, 47,
 48, 128
`\theorem` 94, 95
 theorem 46, 95
`\theorembodyfont` 95
`\theoremheaderfont` 95
`\theoremstyle` 95
 thermodynamic temperature
 26
`\Theta` (Θ) 91
`\theta` (θ) 90
`\thinspace` 132
 Thomas Williams 123
`\tilde` (\tilde{a}) 93
 time 26
 Times 137
`\times` (\times) 92
`\tiny` 40
`\title` 19, 20
 title (BIBTEX) 51, 54
 title 19, 20
`\titlepage` 150
 titlepage 44
 .toc (拡張) 16
 tocdepth (カウタ) 22
 Tomas Rokicki 62
 tools 46
`\top` (\top) 94
`\topmargin` 125, 147
`\toprule` 108
`\triangle` (Δ) 94
`\triangleleft` (\triangleleft) 92
`\triangleright` (\triangleright) 92
`\ttfamily` 41
 Tutorial i
`\twocolumn` 129, 139
 twocolumn 44
 twoside 44, 125
`\txfonts` 99, 137
`\typelcm` 137
`\typelec` 137
`\typeface` 39
`\typical` 107
- U**
- `\u` (i) 28
 Unable to open $\langle file \rangle$.pdf 61
`\unboldmath` 99
`\underbrace` 93
`\underline` 93, 131
 underline 27
 Unix 系 OS 5, 60, 118
 —でのプレビュー 60
 —の基本ツル 9
`\unlhd` (\triangleleft) 94
`\unrhd` (\triangleright) 94
`\unsrt` 53
`\Uparrow` (\Uparrow) 85, 93
`\uparrow` (\uparrow) 85, 93
`\Updownarrow` (\Updownarrow) 85, 93
`\updownarrow` (\updownarrow) 85, 93
`\uplus` (\uplus) 92
`\Upsilon` (Υ) 91
`\upsilon` (υ) 90
 URL 136
`\url` 56, 136
 url 56, 136
`\urlstyle` 136
`\usepackage` 17
 UTF 138
- V**
- `\v` (\check{a}) 28
`\value` 73
`\varepsilon` (ε) 91
`\varepsilon` (varioef) 46
`\varphi` (φ) 91
`\varpi` (ϖ) 91
`\varrho` (ϱ) 91
`\varsigma` (ς) 91
`\vartheta` (ϑ) 91
`\vdash` (\vdash) 92
`\vdots` (\vdots) 94
`\vec` (\vec{a}) 93
`\vee` (\vee) 92
`\verb` 29
 verbatim 46
 verbatim 29
 verbatim 環境 29, 46
`\Vert` (\parallel) 85
`\vert` (\lvert) 85
 Victor Eijkhout 29
 Vine Linux 18
 —への導入 4
`\vline` 88, 106
`\voffset` 125, 147
 volume (BIBTEX) 54
`\vphantom` 100
`\vskip` 150
`\vspace` 127
`\vspace*` 127, 133
- W**
- Warning 17
`\wedge` (\wedge) 92
white space 28
`\widehat` 93
`\widetilde` 93
William Chia-Wei Cheng 120
Windows 6
 —でのダセット 7
 —での日暮 6
 —でのルイ 操作 8
 —でのプレビュー 60
 —のdvips 62
 —の執環境 18
 —のプレビュー 12
 —への導入 4
 —ヘルプ 8
WinShell 18
Word 137
word 19
 —space 34
`\wp` (\wp) 94
`\wr` (\wr) 92
WWW 56
WYSIWYG 3
- X**
- xdvi 12
XEmacs 5
XeTeX 137
Xfig 120
XHTML 63
`\Xi` (Ξ) 91
`\xi` (ξ) 90
Xpdf 60
xr 46
`\xspace` 136
xspace 46, 136
Xy-pic 124
XeTeX 123
- Y**
- YaTeX 18
year (BIBTEX) 51, 54
yomi (BIBTEX) 51, 55
You can't use 'macro
 parameter
 character #' 10
Young U. Ryu 137
- Z**
- `\zeta` (ζ) 90
- あ**
- 空き 33, 132
 垂直 の— 133
 水平 の— 132

単關 の—— 33
 文間の—— 33
 文關 の—— 33
 アキヒ 28
 アクセント 28
 大きい—— 93
 数域 の—— 93
 小さい—— 93
 文中の—— 28
 アクセント記号 28, 137
 アクチフ 文字 68
 アットヤ 50, 67
 阿部 110
 阿部 4
 アンギ 82
 アンド 8, 10
 アンズ 26
 い
 意圖 39
 イタリック体 41
 一級 44
 一級 3
 イニシャルコマンド 14
 入れ子 35
 印刷 53
 ——できる?イ 7
 印刷 60
 印刷 44
 インチ 25
 引用 15, 47
 ——の引用 30
 雑誌名の—— 31
 書籍 の—— 31
 単語の—— 30
 段落の—— 30
 複製 の—— 30
 文の—— 30
 本の名前の—— 31
 引籍 15, 31, 34
 欧文の—— 15
 和文の—— 15
 う
 ウェーブ 28
 ウェブザラ 60
 ウェブペ 56
 受け 27
 内閣憲 60
 ウムヤ 28
 梅嬌 127
 浦歴 110
 上付き 82
 上付き文字 68
 え
 英数 6

英案 68
 エクスプレ 113
 江口慶 4
 エスデ 文字 68
 閲覧 56
 エラ
 ——に対するヘルプ 12
 Label 'key' multiply
 defined 76
 Label(s) may have
 changed. Return
 to get
 cross-ferences
 right. 76
 Misplaced alignment
 tab character & 10
 Missing \$ inserted 10
 Reference 'key' on
 page n undefined ..
 76
 Unable to open
 <file>.pdf 61
 You can't use 'macro
 parameter
 character #' 10
 エラツゼ 7
 円記 14
 演算子 92
 ——の否定 98
 2 項— 83, 92
 大型 92
 お
 欧文
 ——の引籍 15
 大獺子 92
 大沢英一 56
 大翻雄 60
 大翻 iii
 オトマトン 124
 大野 124
 尾関 123
 奥曆 i, 4, 44, 54, 138
 起こし 27
 オゴネク 28
 乙部己 4
 オプション
 10pt 25, 44, 45
 11pt 44, 45
 12pt 44, 45
 12Q 45
 14pt 45
 14Q 45
 17pt 45
 20pt 45
 21pt 45

25pt 45
 30pt 45
 36pt 45
 43pt 45
 9pt 45
 a4j 44, 45
 a4paper 44, 45
 a4var 45
 a5j 44, 45
 a5paper 44, 45
 a6paper 45
 b4j 44, 45
 b4paper 44, 45
 b5j 44, 45
 b5paper 44, 45
 b5var 45
 draft 44, 122
 dvipdfm 112
 dvipdfmx 112, 113, 118
 dvips 112, 113, 118
 english 45, 138
 final 44, 122
 fleqn 44, 78
 landscape 44
 leqno 44
 letterpaper 44
 nosort 57
 nospace 57
 notitlepage 44
 onecolumn 44
 oneside 44, 125
 openany 44
 openright 44
 papersize 45
 report 45, 138
 space 57
 T1 28
 titlepage 44
 twocolumn 44
 twoside 44, 125
 親カウタ 95
 親デレクトリ 9
 か
 会議 54, 56
 改行 32
 ——された表示 11
 ——の抑制 34
 ——を許さない空き 132
 ——を許す空き 132
 array 環境での—— 87
 eqnarray 環境での—— 79
 L^AT_EX での—— 28
 URL の—— 136
 行揃えにおける—— 38
 行末案 としての—— 68

- 原稿における—— 24
 出綴 における—— 24
 数式モード 中の—— 77
 ハオン 途中の—— 32
 複綴 の—— 80
 文章 の—— 32
 連続した—— 28
 会綴 の論文 53
 書いたまま出力する 29
 改綴 32
 改丁 127
 回転 116
 表の—— 117
 文廻 の—— 116
 概要 23
 回路 124
 会証 15
 カウタ 71
 ——の新設 72
 ——の設定 72
 chapter 72
 enumi 72
 enumii 72
 enumiii 72
 enumiv 72
 equation 72
 figure 72
 footnote 72
 mpfootnote 72
 page 72
 paragraph 72
 part 72
 secnumdepth 22
 section 72
 subparagraph 72
 subsection 72
 subsubsection 72
 table 72
 tocdepth 22
 親一 95
 化学
 ——綴 123
 ——式 123
 可換図 102
 かぎ括弧 15, 27, 30, 31
 書き出し 119
 学綴 19
 学会誌 56
 角括弧 14, 15
 ——で囲まれた引数 14
 ——の数式での出力 101
 拡大 116
 区切り記号の—— 86
 図の—— 114
 文廻 の—— 116
 拡張 15, 43
 .aux 16
 .bb 16
 .bbl 16
 .bib 16
 .blg 16
 .bmp 16
 .bst 16
 .clo 15
 .cls 15
 .dtx 15
 .dvi 16
 .eps 16
 .fd 15
 .ins 15
 .jpg 16
 .lof 16
 .log 16
 .lot 16
 .mp 16
 .png 16
 .sty 15
 .tex 16
 .toc 16
 確認 6
 箇条書き 35
 記付 きの—— 35
 説脚 きの—— 35
 番付 きの—— 35
 数値の代入 25
 下線 27, 131
 ——アクセント 28
 ——の意味 27
 画像
 ——の回転 114
 ——の拡大 114
 ——の切り抜き 115
 ——のクリッピング 115
 ——の原標 113
 ——の彖 情報 61
 ——の高さ 114
 ——の縦横 115
 ——のトリミング 115
 ——のびざぐボックス
 ス 114
 ——の幅 114
 ——の張り込み 111
 ——の品質 61
 ——ベクトル 16
 ——への文字の追加 122
 ——編 18
 可綴 16
 単一ペ○ —— 59
 浮動 としての—— 104
 フルカラー—— 16
 無綴 16
 仮想プリンタ 119
 括弧 14
 ——のある行列 46
 ——の大きさの調整 85
 ——の足りない状態 12
 引用のための—— 31
 区切り記号の—— 84
 書籍 のための—— 31
 数式モード 中の—— 78
 スコ○ —— 69
 場合 けの—— 89
 分数での—— 100
 合字 33
 AE の—— 28
 OE の—— 28
 カテゴリ群 67
 下点アクセント 28
 かな漢変換 プログラム 6
 カレントディレクトリ 9
 巻 54
 環境 14, 64
 abstract 23
 appendix 134
 array 46, 86, 105, 109
 cases 101
 center 38
 comment 29
 description 35
 displaymath 78
 document 14
 enumerate 35, 46
 eqnarray 73, 79
 eqnarray* 79
 equation 79
 figure 104
 figure* 129
 flushleft 38
 flushright 38
 itemize 35
 math 78
 matrix 88
 minipage 121, 131
 picture 104, 111
 pmatrix 88
 quotation 30
 quote 30
 spacing 25
 tabbing 105
 table 104
 table* 129
 tabular 46, 104, 105, 109
 tabular* 46
 thebibliography 16, 47,
 48, 128
 verbatim 29, 46
 定理 の—— 95
 問題 の—— 95

例題 の— 95
 関係 92
 間違いの修正 42
 カンデラ 26
 巻末 47, 51

き

キトップ ii
 キ罫 ii
 —からの入力 ii
 ギガ 26
 記号 63
 —の意味 2
 —の重ね合わせ 98
 —の使い方 2
 —の積み重ね 98
 —の分類 63
 —の前の空白 2
 2項罫子 92
 アクセント 93
 アクセント— 28
 円— 14
 演算子 92
 大罫子 92
 大きさが可変の— 83
 箇条書きの— 35
 関係 92
 ギリシャ文字 90
 区切り— 84
 校正 42
 根号 99
 数学 77, 90
 積分 83
 添え字における— 84
 横棒の— 31
 単位 26
 特殊な— 28, 64
 特別な— 8
 文を区切る— 19
 偏微分 97
 矢印 93
 ルト— 99
 記事の検索 21
 奇数 こし 127
 木罫 48
 疑問 11
 脚注 27, 40, 136
 2段組での— 46
 minipage 環境での— 131
 表中の— 107
 行頭
 —の字下げ 24
 行送り 25, 147
 行間 34
 行揃え 38

強調 2, 27, 40
 —の中の単位 26
 見出しの— 42
 文罫 の— 27
 和文の— 27, 42
 行末 132
 —のコンマ 51
 —案 68
 行列 86
 —中の小罫 46
 —に付ける括弧 46
 —の罫線 46
 —の幅 87
 括弧 きの— 88
 局罫 69
 ギリシャ文字 90, 99
 —の大罫 90
 —の小罫 90
 —の変罫 91
 キロ 26
 キログラム 26
 罫線 42

く

空行 28, 77
 空白 68
 —の制御 127
 —の挿入 132
 —の調整 81
 数値と単位の— 26
 行間の— 33
 原稿中の— 28
 コマンドの後の— 136
 四登 き 34
 数式モード 中の— 77
 正しい量の— 33
 ダッシュ間の— 32
 単語 132
 段落 の— 33
 著者 の— 51
 ハオン 間の— 32
 区切り 84
 —罫 84
 項目の— 57
 単語の— 31
 文の— 30
 ページ — 127
 くさび 28
 句読点 19, 26
 国 66
 組版 1, 2, 5
 クラス 15, 43, 44
 —ファール 15
 article 21, 43, 139
 funthesis 144
 jarticle 17, 21, 43, 46, 139

jbook 44
 jclassess 44
 jreport 44, 46, 144
 jsarticle 44, 139
 jsbook 44, 144, 146
 jsclassess 44, 45, 133
 jspj 44
 proc 43
 report 43
 slides 43
 欧風 の— 43
 学会誌 の— 44
 小規模な文書 の— 43
 書籍 の— 44
 標榜 な— 43
 報書 の— 44
 クラスオプション 44
 グルピング 64, 77, 78
 グルデ
 —の終わり 68
 —の開始 68
 グレデ 28

け

警告 11, 17
 Overfull 11
 煩雑な— 11
 計罫罫 137
 罫線 131
 —の太さ 130
 2段組での— 127
 行列の— 87
 表中の— 105
 ケルビン 26
 研究 1
 原稿 5
 —からPDFの作成 60
 —から生成される罫イ
 の種類 7
 —罫 の注意 8
 —罫 の支援 18
 —中の改行 24
 —中の空白 28
 —中の罫 番号 62
 —の校正 42
 —の構成 14
 —の再コンパイル 18
 —の出罫 59
 —の書式 15
 —の処理 3
 —の入罫 8
 —の汎用性 i
 —の先罫分 14
 —のブレイク 16
 —の分割 134
 —の編集 5

—の文字サズ 44
 —の用紙サズ 44
 ドラフト段階の— 44
 原標 113

こ

鈎癖 28
 考察 1
 校正 42
 合線 28
 光度 26
 公立はこだて未来 i
 コド
 カテゴリー— 67
 分類 67
 国際偉 26
 黒線 80
 ゴシック体 42
 コマンド 5, 8, 14
 —の後の空窠 46
 —の情窠 8
 —の引数 14
 —のヘルド 表示 8
 —ラギオプション 9, 62
 イニシャル— 14
 宣言型の— 70
 内部 8
 プリアブル — 14
 命型 の— 70
 コマンドプロンプト ii, 6, 8
 コメント 29
 —アウト 29
 —家 68
 コロン 27
 コンソル ii
 コントロル
 —シヌス 64
 —シンボル 64
 —スペネ 64
 —ワド 64
 コンパル 7
 コンパニオンシリズ 4
 コンピュータ i
 コンマ 17, 27
 全角の— 26

さ

サカムフレックス 28
 サズ 41
 齋藤 138
 再標 12
 索引
 —の見出し 128
 作者 20
 作図 111
 作文 2

—標 48
 雑誌
 和文の— 31
 雑誌名
 —の引用 31
 論誌 53, 54, 55
 —の巻 55
 —の番号 55
 左起 し 44
 参蔵 47, 128
 —データ 47
 —の補題 49
 —の見出し 128
 サンセリフ体 41

し

シェオ 41
 イタリック— 41
 スモルキャピタル — 41
 スラント— 41
 シル ii, 6, 8
 字間 33
 時間 26
 字下げ 19, 23
 —の抑制 24
 行頭の— 24
 段階 めの— 23
 箱の中での— 130
 見出し直後の— 46
 下付き 82
 —家 68
 執筆環境 18
 字詰め 33
 質量 26
 四登 き 34, 42
 紙面
 —構 1
 —の拡張小 25
 シャヅ S 28
 斜線 107
 斜体 3
 住所 66
 修飾子 26
 修蔵 53
 縮小 116
 出版 49, 53, 54, 60
 出版 54
 章 19, 128
 —の見出し 128
 小抵 15
 商蔵 3
 小数 46, 109
 状蔵 124
 章立て 23, 103, 144
 小なり 8
 章標 28

情報 19
 情報システム i
 ショト 28
 書式 15, 43
 原稿の— 15
 書籍 1
 和文の— 31
 書籍
 —の引用 31
 書体 ii, 39
 —の種類 ii
 —の属性 15
 書名 15, 49, 54
 処理
 —の中断 11
 シリズ 41, 49, 54
 ボルド — 41
 ミデムア — 41
 資料の配布 60
 シングルクオト 15, 30
 人名
 Alan Jeffery iii
 Cho Jin-Hwan iii, 61
 Chris Rowley iii
 Colin Kelley 123
 David Carlisle iii, 76, 88,
 109
 Donald Arseneau 57, 136
 Donald Knuth iii, 3, 137
 Frank Mittelbach iii, 94,
 95
 Geoffrey Tobin 25
 Han The Thanh 137
 Hans-Peter Doerr 111
 Harald Harders 82
 Ingo H. de Boer 18
 Jerry Jeichter 109
 Johannes Braams iii
 John Eaton 124
 Karl Berry 62
 Kristoffer Rose 124
 Laurent Siebenmann 122
 Leslie Lamport iii, 3, 4
 Mark Wicks iii, 61
 Michael Downes iii
 Nelson Beebe iii
 Oren Patashnik iii, 47
 Pehong Chen iii
 Piet Van Oostrum 109
 Rainer Schöpf iii
 Raymond Sérout 122
 Richard Koch 18
 Ross Moore 124
 Sebastian Rahtz iii
 Simon Fear 108

- Steven Douglas Cochran
121
Thomas Williams ... 123
Tomas Rokicki 62
Victor Eijkhout 29
William Chia-Wei Cheng
120
Young U. Ryu 137
阿部 110
阿部 4
内藤憲 60
梅藤 127
浦藤 110
江口康 4
大沢英一 56
大鶴雄 60
大鶴 iii
大野 124
尾関 123
奥藤 i, 4, 44, 54, 138
乙藤己 4
齋藤 138
竹野滋 124
中川仁 18
中賢 iii, 4
平藤 iii
広瀬 18
藤田眞 4, 123
山賀人 123
渡邉 iv
- す**
図 128
—の中揃え 103
—の見出し 128
—揃 128
数罫 83
数罫 77, 90
数式
—中の角括弧 101
—中の空白の調節 81
—中のテキスト 80
—中の波弧 101
—中の太字 46
—中の丸弧 101
—の位置 44
—の組版 77
—の書体の変更 80
—中の文章 81
—の左揃え 78
—の表罫の調整 89
—の太字 99
—番の位置 44
—モード 77
番付 ぎの 79
複数 の番付 ぎ 79
- 文中 77
別行 78
数式モード 77
スコデ 69
変数の 69
スタイル
参観 16
文罫 の 47
スタイル 15
図表
—の配置 104
—揃し 103
図揃し 103
スモルキャピタル 体 41
図揃 22, 128
スラッシュ付き O 28
スラント体 41
- せ**
制御り 64
成形 5
—ファール 7
整形 5
製本 53
整列 109
積罫 83
セクション 28
節 19
絶的 な長さ 25
接頭 26
セデヲ 28
セミコロン 27
全角ダン 31
全角 6
宣言 64
—型のコマンド 70
センチメートル 25
- そ**
相懸 71
—できるもの 71
—に関わる警告 76
—に必要なヲル 7
—の工夫 73
—の仕組み 71
—のラベロ 表示 46
別の文書との 46
装置 45
添え字 82
上付きの 82
下付きの 82
ソス 16
ソスヲイ 5, 7, 16, 43
—の編集 5
- た**
ダン
- 全角 31
波 31
倍角 31
タミナル 6, 7, 8
タイ 28
第 128
ダイグラム 124
大罫 69
大学 54
大弧 15
大なり 8
タオセット 5
—後の生成ファール 7
—の終了 12
—の続行 12
—の続行 12
ノンストップの 12
タオヲ 体 41
題名 20, 55
ダヤグラム 102
ダガー 28
高さ
x の字の 25
ヘッダの 125
本文罫 の 127
ルヲ 99
竹野滋 124
多言罫 138
多段組 46, 128
ダッシュ 31
—の両隣 32
縦罫 107
縦棒 8
ダブルクオット 27, 30
ダブルクリック 6
ダブルスペース 25
ダブルダガー 28
単位 25, 26
—のずれ 25
—の接頭 26
段組 44
短符 28
短罫 28
単語 19
—間の空白 33
—の引用 30
段標 28
段落 19
—間の空白 33
—の引用 30
- ち**
地域 66
中揃え 38
中弧 15
注記 49
注釈 19, 27

—の位置 27
 —の役割 27
 中途ファル 7
 長符 28
 著權 47
 著物 47
 著者 54, 55
 —の頭字 51
 —の姓名のあいだ ... 51
 —の統一 49
 —の読み 55
 チルダ 8, 28

て

ディエシス 28
 定数 90, 91
 定理 46
 デレクトリ 8
 —の移動 8
 —の新機 8
 —の作り方 8
 親— 9
 カレント— 9
 テキストタタ ii, 5, 6
 Emacs 6
 メモ帳 6
 テキストフルイ 5
 テキストモード 77
 デバネ 45
 —ドラボ 17, 138
 手書 i
 テラ 26
 点 94
 —の不在 i 28
 —の不在 j 28
 句読 26
 転載 47
 電機 60
 電流 26

と

通し番号 21
 ドキュメントクラス 43
 —オプション 15, 17
 独性 1
 読者 47
 特機 29
 特機 28
 ドラフト 44
 ドル 8, 10

な

中川仁 18
 長さ 26
 —の単位 25
 1列の— 87

絶的 な— 25
 中賢 iii, 4
 ナノ 26
 波符 28
 波抵 8, 15
 —で囲まれた引数 ... 14
 —の数式での出力 .. 101
 —の役割 14
 数式モード 中の— ... 78
 スコ~~ラ~~のための — ... 69
 文献リスト中の— ... 50
 波ダシ 31
 並び替え 50
 引順 の— 53
 文献の— 50
 難解な用語 19
 ナンバー 8, 10

に

二重 15
 二重かぎ括弧 15, 31
 二重 31
 二重 28
 二重 43, 44
 日語 6
 —丸 6
 —の幅 25
 日語 6
 入力 6
 英数 6
 全角 6
 日語 6
 半角 6
 入換 9
 入通 りの文字の出力 29
 任意数 14
 認機 108

ね

熱劣機 26

の

ノンストップ 12

は

場合 け 87, 101
 バジヨン 情報 7
 パセント 8, 29
 パセン 15, 27
 倍角ダシ 31
 媒体 39
 ハ~~ボ~~ンク 60
 ハオ~~キ~~ヨン 31
 ハオン 31
 —の両隣 32
 配列

—の要素の区切り ... 68
 バウ~~キ~~グボックス 61, 113,
 114, 120
 博識 53
 箱 129
 広範 な— 130
 ハチェック 28
 バッククラウド 6, 12
 バックスラッシュ 8, 10, 14
 パッケージ 15, 43
 —オプション 17
 abstract 139
 afterpage 46
 amsbsy 99
 amsfonts 94
 amsmath 45
 amssymb 80, 94, 120
 amsthm 95
 array 46
 babel 45
 balance 129
 bm 46, 99
 booktabs 107, 108
 calc 46
 cite 57
 color 45
 comment 29
 dcolumn 46, 109
 delarray 46, 88
 eepic 111, 120
 enumerate 46
 epic 111, 120
 exceltex 111
 fontenc 28
 ftnright 46
 geometry 127
 graphicx 17, 45
 hline 46
 indentfirst 24, 46, 149
 jarticle 138
 jbook 138
 jreport 138
 jsarticle 138
 jsbook 138
 jsclasses 4, 138
 labelfig 122
 latexsym 94
 layout 46, 125
 leftidx 82
 longtable 46
 morisawa 138
 multicol 46
 multirow 109
 okumacro 31, 138
 okuverb 138
 OTF 138

- pict2e 111
 PStricks 120
 pxfonts 99, 138
 setspace 25
 showkeys 46, 76
 subfigure 121
 tabularx 46
 textcomp 26
 theorem 46, 95
 tools 46
 txfonts 99, 137
 type1cm 137
 typelec 137
 url 56, 136
 UTF 138
 varioref 46
 verbatim 46
 xr 46
 xspace 46, 136
 発行月 54
 発符 49, 54
 バッチ処理 3
 バッチモード 12
 ハット 8, 82
 発表年 55
 幅
 1 行の— 147
 M の字の— 25
 画像の— 114
 行列の— 87
 参蔵 リストの— 48
 字下げの— 24
 箱の— 129
 表の— 46
 傍注の— 127
 本文の— 127
 文字の— 41
 要素の— 100
 パラグラフ 28
 パラメタ 25
 — 案 68
 版 5, 54
 半靱 8
 半角 6
 番号
 — の深さ 22
 学籍 150
 箇条書きの— 35
 脚注の— 107
 論誌 の— 54
 参蔵 の— 49
 数式の— 79
 図の通し— 104
 図表の通し— 104
 図誌 しの通し— .. 72
 通し— 21
 表の通し— 104
 複靱 の数式の— 79
 見出しの通し— 72
 番付 き数式 79
 版面 125
 — の設定 127
 ひ
 引数 9, 14
 角括弧で挟まれた— .. 14
 コマンドの— 14
 波軀 で挟まれた— .. 14
 任意の— 14
 ハオで 始まる— 9
 必須の— 14
 ピコ 26
 非常に長い表 46
 左揃え 38
 数式の— 78
 日付 20
 必類 14
 ビットマップ画像
 可縊 16
 フルカラー— 16
 無縊 16
 否定
 演算子の— 92
 表 128
 — 中の小数 46
 — の罫線 46
 — の中揃 え 103
 — の見出し 128
 — 扱 128
 長い— 46
 べを 跨ぐ— 46
 秒 26
 表組み 103, 107
 欧文の— 103
 表縊 103
 表紙の作成 150
 表題 19, 53, 54
 — の役割 20
 同ペを — 44
 独立ペを — 44
 表紙の— 150
 表掲 し 103
 表扱 22, 128
 平靱 iii
 ピリオド 27, 109
 全角の— 26
 広瀬 18
 ふ
 部 19, 128
 — の見出し 128
 ファイル
 — の情報 8
 — の名前の変更 8
 config.pdf 62
 config.ps 62
 DVI— 7
 dvipdfmx.def 112
 gnu-head.pdf 115
 jplain.bst 52
 ltboxes.dtx 129
 Makefile 18
 Rplots.pdf 120
 印刷できる— 7
 画像 104
 組縊 の— 7
 クラス— 125
 ソス — 5
 ドキュメントクラス— .. 15
 文献スタイル — 53
 文献データ — 50
 文書クラス— 15, 78
 ログ— 7
 ファネダ 113
 フキサ 41
 サンセリフ— 41
 タイタ 41
 ロマン — 41
 フォルダ 8
 — の移動 8
 — の新縊 8
 — の作り方 8
 フォント
 — のアウヰラ 化 .. 113
 — の大きさ 41
 Type1 形式の— 62
 付縊 54
 複靱 79
 複縊 き数式 79
 複縊 の引用 30
 藤田儀 4, 123
 伏字 L 28
 物理量 26
 符点 28
 浮動 104
 — の位置 104
 太字 99
 数或 の— 46
 ブラックボをボ 体 .. 80
 プリアブル 17, 51
 — コマンド 14
 フリマウよア 財団 2
 プレビュー 3, 6, 120
 Red Hat での— 12
 Unix 系 OS での— .. 12
 Windows での— 12
 プレビューア 12
 付録 128

—の見出し 128
 付録の追加 134
 プログラミング言語 3
 プログラム
 Acrobat Reader 61
 Adobe Acrobat . 119, 121
 Adobe Reader 60, 61
 Aleph 138
 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 45
 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 45
 $\mathcal{B}\mathcal{B}\mathcal{T}\mathcal{E}\mathcal{X}$ 47
 Calc 110
 $\text{Calc}2\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 110
 cd 8
 copy 8
 cp 8
 CreateBB 16
 del 8
 dir 8
 Dviout 12
 Dvipdfm 16, 61
 Dvipdfmx 61
 dvipsk 62
 EasyPackage 4
 Easy $\mathcal{T}\mathcal{E}\mathcal{X}$ 18
 ebb 16, 113
 Emacs 18
 epstopdf 117
 $\epsilon\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 137
 Excel 110, 111
 Exel2tabular 110
 file 113
 Foxit Reader 60
 Ghostscript 60
 Gnuplot 18, 123
 Grapher 120
 help 8
 Hyper $\mathcal{T}\mathcal{E}\mathcal{X}$ 61
 identify 113
 Illustrator 119, 120
 info 9
 j $\mathcal{B}\mathcal{B}\mathcal{T}\mathcal{E}\mathcal{X}$ 47, 50
 Keynotes 120
 Lambda 138
 Lamed 138
 $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 3
 latex 6
 latexmk 52
 less 9
 ls 8
 MacOS X WorkShop .. 4
 Make 18, 52
 man 8
 Mathematica 120
 MATLAB 120, 121
 METAFONT 120

METAPOST 16, 120
 Microsoft Office 3
 mkdir 8
 move 8
 mv 8
 Mxdvi 60
 Octave 120, 124
 Omega 138
 OmniGraffle 120
 OpenOffice.org 3
 Pages 120
 $\text{PDF}\epsilon\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 137
 $\text{PDF}\epsilon\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 137
 $\text{PDF}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 60, 137
 $\text{PDF}\mathcal{T}\mathcal{E}\mathcal{X}$ 137
 pdvips 62
 Perl 111
 Photoshop 120
 $\text{Pic}\mathcal{T}\mathcal{E}\mathcal{X}$ 120
 platex 6
 PostScript 60
 PrimoPDF 119
 ps2jpdf 117
 ps2pdf 117
 pxdvi 12
 R 120
 rm 8
 SciLab 124
 Susie 112
 $\mathcal{T}\mathcal{E}\mathcal{X}$ 3
 $\mathcal{T}\mathcal{E}\mathcal{X}\text{Shop}$ 18
 Tgif 18, 120
 Vine Linux 18
 WinShell 18
 Word 137
 xdvi 12
 $\mathcal{X}\mathcal{E}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 137
 Xfig 120
 Xpdf 60
 $\mathcal{X}\mathcal{Y}\text{-pic}$ 124
 $\mathcal{X}\mathcal{M}\mathcal{T}\mathcal{E}\mathcal{X}$ 123
 YaTeX 18
 エクスプレット 113
 ファクタ 113
 プレビュー 120
 プログラムリスト 29
 文 19
 —の引用 30
 分符 28
 文間空 33
 文献
 —の探し方 i
 文献
 —の簡略 53
 文献スタイル
 abbrev 53

alpha 53
 jabbrv 53
 jalpha 53
 jplain 53
 junsrt 53
 plain 53
 unsrt 53
 文献データ 50
 文献の種類
 article 53
 book 53, 54
 booklet 53
 inbook 53
 incollection 53
 inproceedings ... 53, 56
 manual 53
 masterthesis 53
 misc 53
 phdthesis 53
 文書 19
 — 彙 1
 —の概略 21
 —の正確 2
 —のスタイル 46
 —の末尾 48
 組織 の— 6
 でたらめな— 1
 ビジネス— 38
 マップ 言語による—
 3
 ワザロソフによる — . 3
 文章
 — 現 2
 文書クラス 43
 — オプション 17
 分数 84
 —の書き方 89
 連— 89
 文献 77
 へ
 米学会 45
 ペジ
 — 語 16, 59
 —の行数 127
 —の区切り 127
 —の最部 104
 —の最部 104
 —の先頭での空き .. 127
 —の末尾での空き .. 127
 —のような箱 131
 —の余白 125
 —レイアウト 125
 改— 127
 表題 44
 ベクトル

- 罫 92
 ベクトル画像 16
 べた書き 29
 別裁 77
 ヘルプ
 エラに 対する—— 12
 簡易の—— 9
 詳細な—— 9
 編集 5
 原稿の—— 5
 ソース 5
 編集者 54
 変数 25, 69, 91
 ——の有無 69
 変換 90
 偏微分 97
- ほ**
- ポット 25
 法 83
 傍注 27
 ボルド 体 41
 補綴 の追加 19
 補綴 15
 ポンド 28
 本の名前の引用 31
- ま**
- マチ 1, 39, 40, 43, 60
 マチロ 26
 マウス 31
 まえがき 23
 前書き部分 17
 マクロ 15, 43
 ——の再編 65
 ——の作成 97
 ——の定義 65
 ——パッケージ 43
 数値 の—— 45
 パッケージ 化された—— 15
 マクロパッケージ 15
 マニュアル 53
 ——配布 60
 丸括 15, 27
 ——の数式での出力 .. 101
- み**
- 右揃え 38
 見出し 19, 21
 ——の作成 21
 ——の直後 46
 ——の通し番号 22
 ——の深さ 21
 ——の変更 128
- 目次 の—— 21
 見出し一覧 22
 ミディアム 体 41
 ミリ 26
 ミリメートル 25
 明朝体 42
- む**
- 無効 68
 無視される文字 68
- め**
- 命令 14, 64
 ——型のコマンド 70
 メートル 26
 メガ 26
 メモ帳 5, 6
- も**
- モド 77
 数式 77
 テキスト—— 77
 目次 19, 22, 128
 ——の作成 71
 ——の番号 けの深さ .. 22
 ——の深さ 22
 ——の見出し 128
 ——の見出しの変更 .. 128
 ——用の中途ファイル .. 16
 ——用の見出し 21
 secnumdepth 22
 tocdepth 22
 図—— 22, 128
 表—— 22, 128
 文字 2, 19, 39
 ——間置 33
 ——サズ 44
 文字サズ 45
 文庫 138
 モル 26
- や**
- 矢印 93
 山置人 123
 山括 15
- ゆ**
- ユニコード 138
- よ**
- 揚符 28
 用語 19
 ——の統一 135
- 用紙 125
 ——の空白 125
 ——の大きさ 44
 ——の大きさの指定 61, 62
 ——の罫 44
 ——の方向 44
 B 列の—— 62
 欧標 の—— 62
 用紙サズ 45
 ヨロッパ 語圏 137
 抑揚符 28
 横線 88, 106
 余白 120, 125
 読み 55
 読み方 3
 予約案 64
- ら**
- ラベル 71
- り**
- 利用 2
 リング 28
- れ**
- 列籍
 行列における—— 87
 小数 を揃える—— 109
 表中の—— 105
 レポート 作成 45
 連数 89
- ろ**
- ロマン 体 41
 ログフォル 7, 16
 ロング 28
 ——ハンガリア 28
- 論文
 —— 48
 ——における図表 106
 会議 の—— 53
 科学 i
 修士 53
 博士 53
- わ**
- ワチソフト 3
 枠
 ——と文字の間隔 130
 ——の太さ 130
 渡 iv
 和文
 ——の引籍 15

L^AT_EX による論議 の手引き 改訂 3 版

© FUNNIST 2003, 2004, 2006

© 渡邉 2003, 2004, 2005, 2006

発行	2003 年 11 月 第 1 版発行 2004 年 11 月 第 2 版発行 2006 年 3 月 第 3 版発行
----	--

編集	FUNNIST
監修	<small>わたなべとおる</small> 渡辺徹 (thor@tex.dante.jp)
装丁	<small>きむらけんいち</small> 木村健一
サポート	http://tex.dante.jp/typo/

公立はこだて未来 FUNNIST 編集員

Future University-Hakodate Network and Information System Tutorial Committee