

L^AT_EX 2_εによる論文作成の手引

松田 七美男著
濱田 龍義編集

目次

第 I 部 初級コース	1
1 では ”こんにちはラテフ!” から	2
1.1 ソースを emacs で編集	2
1.2 platex(ぴーらてふ) で DVI ファイルへ	2
1.3 xdvi によりレイアウト確認	3
1.4 まとめ	4
1.4.1 L ^A T _E X 2 _ε で整形された文書を作成する手順	4
1.4.2 最小限度記述が必要なコマンド 3 行	4
2 章・節建て	6
1 セクション	6
1.1 サブセクション	6
1.1.1 サブサブセクション	6
2 次のセクション	6
2.1 サブセクション	6
2.2 次のサブセクション	6
1 はじめに	7
2 実験	7
2.1 装置	7
2.2 結果	7
1 はじめに	7
2 実験	7
2.1 装置	7
2.2 結果	7
3 タイトルと著者	8
4 目次	9
5 ヘッダやフッタ	10
5.1 形式	10
5.2 ページ番号	10
6 数式	12
6.1 数式モード	12
6.1.1 eqnarray 環境	12
6.2 数式フォントと空白の扱い	13
6.2.1 英数字と空白	13
6.2.2 立体文字	14
6.2.3 ギリシャ文字	14

6.3	数式記法	16
6.3.1	上付き・下付き文字	16
6.3.2	積分・級数和	16
6.3.3	根号	16
6.3.4	分数	17
6.3.5	行列	17
7	図の貼り込み	19
7.1	図を入れる空白を用意しておき、後に糊で貼る	19
7.2	EPS形式の図を貼り込む	19
8	表	21
8.1	tabular 環境	21
8.1.1	cline	22
8.1.2	multicolumn	23
9	フォント	23
9.1	特殊文字	24
9.2	英数字の書体	24
9.3	大きさの指定	25
10	箇条書き (List)	26
10.1	マーク付き箇条書き : itemize	26
10.2	番号付き箇条書き : enumerate	27
10.3	見出し語付き箇条書き : description	28
10.4	引用環境 : quote, quotation	29
10.5	文字揃え環境 : flushleft, center, flushright	29
10.6	ミニページ : minipage	31
11	空白制御	32
11.1	垂直方向の空白	32
11.1.1	空白を入れる命令	32
11.1.2	行間隔を制御する命令	33
11.2	水平方向の空白	34
12	枠囲い	36
12.1	箱 (ボックス) - box -	36
12.1.1	単純な箱 : fbox	36
12.1.2	少し自由な箱 : framebox	37
12.1.3	段落が入る箱 : parbox	38
13	罫線	39
13.1	黒塗の箱 : rule	39
13.1.1	自動的に長さが調節される罫線 : hrulefill, dotfill	39
14	長い文章の編集	41
14.1	他のファイルの読み込み : input, include, includeonly	41
14.2	専用マクロの読み込み	42

第I部 初級コース

文書作成や整形はワープロソフトしか使ったことがなくて、**L^AT_EX** を使うのは初めてという人のためのコースです。L^AT_EX は、理系の論文作成には最適の文書作成システムであると思いますが、動く環境を準備するのに手間が掛かりますし、マクロ (スタイルファイル) による拡張機能が優れていることが、却って初心者には混乱の元になるようです。従って、なるべく標準の配布セットでできることを中心に説明していきます。

さて、L^AT_EX とは別に、文書を編集するために**エディタ**を使いこなすことが長期的には重要です。UNIX 系では、Emacs が定番です。強力なマクロ機能を使って L^AT_EX 用のモードが構築できます。そのようなモードとしては AUC-TEX というのが世界的には有名ですが、ここでは敢えて『野鳥』(yatex) を使います。日本人 (慶応大学の広瀬さん) が作者であり、ドキュメントが日本語ですし、サポートも迅速という点もさる事ながら、とても使い易いことに感心しました。実は、Windows 上でも秀丸エディタのマクロ『T_EXMAC』が同様の機能を提供してくれます。従って、PC-UNIX はちょっと自分の手に負えないという人も、大丈夫なのです (もっとも、ここまでインストールが易くなった PC-UNIX が駄目と言う人は、TeX も駄目かも知れませんが)。とにかく、自分のお気に入りのエディタを決めて馴染んでおきましょう。『Microsoft Word』をエディタの代わりに使うのはなるべく止めましょう。ワープロソフトはエディタ以外の機能が多すぎて重くコンピュータに負荷がかかります。軽くサクサク動くエディタという分類のソフトを使いましょう。

文字入力に際しては、日本語変換が重要です。この点に関しては Windows 系の方が種類も豊富で機能も高いものが多いのですが、UNIX 系もだいぶ賢くなってきましたし、**ネットワーク上で辞書が共有できる**という利便性は捨てがたいです。このコースでは、PC-UNIX の Linux を標準にしている、日本語入力は定評のある Anthy というソフトウェアを使いますので、以前に比べると日本語の変換効率は期待できます。

謝辞 この文書「L^AT_EX 2_εによる論文作成の手引」は東京電機大学 松田 七美男教授が執筆された「L^AT_EX による論文作成の手引」を元に濱田 龍義が福岡大学の環境に合わせて一部記述をあらため、L^AT_EX 2_εについての教科書に編集しなおしました。この文書のソースファイルをころよく公開していただいた松田教授に感謝致します。

1 では ”こんにちはラテフ!” から

L^AT_EX 2_εで文章を作成整形する大まかな手順を覚えましょう。普通のワープロと違って、印刷時のイメージを見ながら編集というわけにはいきません。少なくとも3つの過程を経なければイメージが出力されないのです。これは初め短所と感じられるでしょうが、実は過程を分ける方が却って速いことが分かってくることと思います。

1.1 ソースを emacs で編集

とにかく簡単な文書を入力してみましょう。

例題 1

エディタで次のテキストを作成して、`samp1.tex` という名前のファイルに保存しなさい。

```
\documentclass[a4j]{jarticle}
\begin{document}
こんにちはラテフ!
\end{document}
```

Linux の X Window System を使います。コンピュータに命令を入力するための端末エミュレータ **GNOME** 端末, **Konsole**, **kterm** などの窓を開いてください。そこから、ファイル名前を引数に取って Emacs を起動します。ここでは数ある Emacs 系のエディタのうち Emacs-21.3 というものを使います。最後に `&` をつけると後で便利です。

```
emacs samp1.tex &
```

入力を終わったら必ず保存 (**Save**) してください、そうしないと他のアプリケーションから読み込むことができません。ついでに一生懸命書いた文章はこまめに保存する癖をつけて置きましょう。いつ事故が起こってファイルが失われるか判らないのです。数分ならまだしも数時間かけた文章を一瞬の内に失う時の喪失感とこまめに保存しておけば良かったという後悔の念を味わった人もいるでしょう。Emacs では `C-x C-s` が保存のコマンドです。

1.2 platex(ぴーらてふ) で DVI ファイルへ

続いてこの平テキストファイル `samp1.tex` から T_EX の独自の **DVI**(DeVice Independent) ファイル `samp1.dvi` ファイルへと変換します。**platex**(ぴーらてふ) というコマンドを使います。

例題 2

次のコマンドを実行して、`samp1.tex` から DVI ファイル `samp1.dvi` を作成しなさい。

```
platex samp1.tex
```

先程 emacs を `&` を付けて起動したので (この場合、emacs はバックグラウンドで動いていると言われます)、同じ端末エミュレータで `platex` を実行させることが可能です。もし `&` を付け忘れた場合は、他のコマンドを実行させることができませんから、ウィンドウマネージャー (**sawfish**, **afterstep**, **fvwm** etc.) のメニューを呼び出して (一般にはルートウィンドウ上でマウスの左ボタンをクリックする)、もうひとつ端末エミュレータを起動し、そこから `platex` を実行させることになります。

とても簡単なテキストなので入力ミスはなかったと思いますが、一般には入力ミスや (入力は正確でも) 文法ミスによりエラーが出て止まってしまう場合が多く、もう一度ソースファイルを見直して修正しなけ

ればなりません。この辺が $\text{T}_{\text{E}}\text{X}$ で不便なところです。まあしかし、数をこなす内にあまり間違いはしなくなりますし、エラーを起こし易い箇所も判るようになりますから。

1.3 xdvi によりレイアウト確認

印刷する前に仕上りを確認しましょう。実は DVI ファイルには印刷先 (周辺機器) の情報が含まれていません、純粹に文字列や図の配列を記述してあるだけです。すなわち、出力機器に依存しないのです。この DVI ファイルの情報を解釈して実際に出力先の機器に見合ったビットマップに展開し、画面に印字したりプリンタに印字したりするのはドライバの役割です。X の画面に出力するドライバを使うアプリケーションが **xdvi** です。印刷の前に確認すると言う意味で、プレビューアとも呼ばれます。

例題 3

次のコマンドを実行して `samp1.dvi` の内容を確認しなさい。

```
xdvi samp1.dvi &
```

`platex` は DVI ファイルを作成すると終了してしまいますから、続けて `xdvi` を起動することができます。ところで、気に入ったレイアウトを完成させるまで、何度でも確認したいので `xdvi` は終了させたくありません。従って、バックグラウンドで動かして `xdvi` を終了しなくても他のコマンド (`platex`) が実行できるようにします。この時点で、`kterm(pxvt)` と `emacs` と `xdvi` が同時に X の画面に表示されている筈です。これは、マルチタスクの UNIX であるから可能なのであって、『あーなんて便利なんだ』と感動してください。

出力例

こんにちはラテフ!

で、結局レイアウトされて現れた結果は、以上のようにになっている筈です。つまらないですか? 最初は大方そんなものです。そこですかさず意地悪な質問をば、

例題 4

上のレイアウト結果で気づいた点を述べなさい。

そう文頭に 1 文字空白が置かれていますね。これは **indent**(字下げ) と呼ばれ、段落の始まりを示しているのです。即ち、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X } 2_{\epsilon}$ は (アメリカ生まれですが) 段落を明確に現すように仕組みられているのです、そのことは逆に書き手には **段落を意識して文章を書くよう要求している** と言ってもいいでしょう。大袈裟かもしれませんが、欧米人の考えるところ論理的な文章の展開においては、段落こそ文章の基本であるといった発想に基づくものではないかと思われまます。論理的な文章を書くのが苦手 (それは単に経験がなかっただけの事だと思いますが) な人は、長い文章を書こうなどと思わず、短くてもいいから、すっきりと、意味のはっきり判る **段落にまとめる訓練** を積むといいと思います。

1.4 まとめ

1.4.1 L^AT_EX 2_εで整形された文書を作成する手順



普通のワープロソフトと異なり複雑に思えるかも知れません。しかし、出版の場合はこのようなあるいはもっと複雑な行程に分割するするのが普通です。つまり高い印字品質を得るには、それぞれの別の行程に分けた方がよいのが現実なのです。L^AT_EX 2_εを旨く活用すれば、そのまま出版物の版下となり得ます。

1.4.2 最小限度記述が必要なコマンド 3行

```
\documentclass{jarticle}
\begin{document}
... 文章内容...
\end{document}
```

例題 1 を振りかえって L^AT_EX 2_εのいわば文法を確認しましょう。この文章の内容は 3 行目の『こんにちはラテフ!』だけです。で、残りの 3 行の部分が最小限記述が必要なコマンドとなっています。これはキマリですから、覚えてください。

練習 1

次のソースの正しくない箇所を指摘しなさい (3 箇所あります)。

```
\documentclass[a4j]{jarticle}
\begin{document}
  先頭の空白は無視される。
\end{document}
```

練習 2

少し長い文章を打ち込んでレイアウト、特に段落の付き方を確認しなさい。

```
\documentclass[a4j]{jarticle}
\begin{document}
  新年早々、風邪で寝正月である。昨年未から、やばいなど感じてはいたのだが、
  十二月二十八日に風呂に入ったとたん、風邪が一気に悪化してしまった。...(略)

  すぐ薬を飲めばすむことだが、私は薬をめったに飲まないで、
  異様なくらいに効いてしまう。胃腸薬や鎮痛剤も、十年以上飲んだことがない。
  とにかく家には薬というものがほとんどないのだ。...(略)

  「こりゃいいぞ」

  と布団にくるまって喜んでいるうちに、今度は背中がゾクゾクしてきた。...(略)

\hfill{(群ようこ「日常生活」より)}
\end{document}
```

レイアウト結果


新年早々、風邪で寝正月である。昨年未から、やばいなど感じてはいたのだが、十二月二十八日に風呂に入ったとたん、風邪が一気に悪化してしまった。...(略)

すぐ薬を飲めばすむことだが、私は薬をめったに飲まないで、異様なくらいに効いてしまう。胃腸薬や鎮痛剤も、十年以上飲んだことがない。とにかく家には薬というものがほとんどないのだ。...(略)

「こりゃいいぞ」

と布団にくるまって喜んでいるうちに、今度は背中がゾクゾクしてきた。...(略)

(群ようこ「日常生活」より)

 最後の`\hfill{文字列}`は、括弧内の文字列を右寄せにして出力してくれるので、重宝します。

練習 3

xdvi のオプションにはどのようなものがあるか調べなさい。UNIX にはオンラインマニュアルが整備されているので、

```
man xdvi
```

で調べることができます。

2 章・節建て

学術論文等はある程度形式が定まっていますから、それにふさわしい仕上りはスタイルで指定しておいて、中身だけに集中できるようにと L^AT_EX 2_ε は考案されました。実際、仕上りに関して、タイトルはゴシックの何ポイントで...、章はゴシックの何ポイントで初めてインデントは幾つで..等は L^AT_EX 2_ε がいいあんばいにレイアウトしてくれます (気にいらない場合は自分で変えることもできます)。また一般にちょっと長い論文では、章節項に分けて番号をふります (目次にも使われる) が、これも L^AT_EX 2_ε が自動的に通し番号を付けてくれます。書き手は、ここから章でこれは節でと言う風に文章の各部分をただ指定していけばいいのです。この指定に使うコマンドがセクション型コマンドです。

```
\part(部), \chapter(章)
\section(節), \subsection(項), \subsubsection(目)
\paragraph(段落), \subparagraph(小段落?)
```

例題 5

次の内容で samp5.tex を編集し platex にかけて、samp5.dvi を作成しましょう。

```
\documentclass[a4j]{jarticle}
\begin{document}
\section{セクション}
\subsection{サブセクション}
\subsubsection{サブサブセクション}
\paragraph{パラグラフ}
\section{次のセクション}
\subsection{サブセクション}
\subsection{次のサブセクション}
\end{document}
```

出力例：章・節建て

1 セクション

1.1 サブセクション

1.1.1 サブサブセクション

パラグラフ

2 次のセクション

2.1 サブセクション

2.2 次のサブセクション

前の段落との間隔を適当に空けて、インデント無しで通し番号を先頭に割り当て、見出し語を表示しています。この見出し語は目次にも使われます。**jarticle** というスタイルでは通し番号は section 番号 [.subsection 番号] [.subsubsection 番号] の様につけられます。従って、paragraph や subparagraph は直接見出し語から始まります。修論で使う可能性のある **jreport** や **jbook** では chapter 番号 [.section 番号] [.subsection 番号] の様につけられます。従って、subsubsection 以下が直接見出し語から始まります。

練習 4

空白や空行に注意して、次の2つの文章をそれぞれ編集し、レイアウトの相違を確かめましょう。

```
\documentclass[a4j]{jarticle}
\begin{document}
\section{はじめに}
なるべく行間を詰めて
\section{実験}
全く空行を入れないで
\subsection{装置}
書いていますからとても見にくいですが
\subsection{結果}
この様にきちんとレイアウトされます
\end{document}
```

```
\documentclass[a4j]{jarticle}
\begin{document}
\section{はじめに}
今度は空行を適当にいます
\section{実験}
従って、見やすいソースとなります
\subsection{装置}
調子によって余分な空白空行も入れました
\subsection{結果}
が、隣と変わらないレイアウトになります
\end{document}
```

セクション型コマンドの前後の空白空行

1 はじめに

なるべく行間を詰めて

1 はじめに

今度は空行を適当にいます

2 実験

全く空行を入れないで

2 実験

従って、見やすいソースとなります

2.1 装置

書いていますからとても見にくいですが

2.1 装置


調子によって余分な空白空行も入れました

2.2 結果

この様にきちんとレイアウトされます

2.2 結果

が、隣と変わらないレイアウトになります

 セクション型コマンドの前後の空白や空行はレイアウトに影響を与えないことが解りました。従って、適当に空白や空行を入れて読み易いソースファイルを書くように心掛けましょう。ただし、段落の開始・終了に関しては空行が意味を持ちますから注意してください。

3 タイトルと著者

文章にはタイトルと著者名が付きますが、例によって見栄えを良くするためにいろいろレイアウトを設定しなければなりません。L^AT_EX 2_εではもちろんそのためのコマンドを用意しています。

`\title{...}`, `\author{...}`, `\date{...}` で中身を指定しておきます。

`\maketitle` を置いた場所以降のページの先頭からをしたにタイトルと著者と日付がレイアウトされます。

例題 6

次の内容で `samp6.tex` を編集して、レイアウトを確かめましょう。

```
\documentclass[a4j]{jarticle}
\title{超電導薄膜の作成技術：最近の展望}
\author{松田彩}
\date{平成 8 年 10 月 22 日}
\begin{document}
\maketitle
\section{はじめに}
最近、超電導薄膜の作成技術の進歩に応じて、その産業における利用への期待は急速に高まっています。....
\end{document}
```

`date` にはコンパイルをした日付が自動的に設定されます。したがって、日付をいれたくない場合には `\date{}` と中身を無しにします。この中央揃えのレイアウトが気に入らない場合には自分でレイアウトします。その方法は、中級コースになってからということで、暫くはこの L^AT_EX 2_ε 既定のものを使いましょう。

練習 5

中央揃えはまあいいでしょうが、タイトルの文字が弱いし、著者名が小さいですね。どうしても変えたいという人のために、簡単な方法を紹介します。{ } の中でフォントの大きさと種類を指定すればいいのです。

```
\titlef{\LARGE\bf 超電導薄膜の作成技術：最近の展望}
\authorf{\LARGE 松田彩}
```

としてみても、確かめてください。`\bf` は太字の指定です。大きさの指定は地の文の文字に対する相対的な大きさ (=縦横の拡大率) で定義されます。`\Large` は約 1.4 倍 (正確には 1.2² 倍)、`\LARGE` は約 1.7 倍 (正確には 1.2³ 倍) の大きさのフォントが指定されます。大きさ 種類 の順に指定してください。`{\bf\LARGE ...}` とすると `\bf` 指定が有効になりません。

4 目次

章節の番号と表題を一覧にした目次は、修士論文には必要ですが、自分で作成するのはとても骨が折れます。L^AT_EX 2_εでは目次の自動作成機能があり、これだけでも使ってみる価値があります。コマンドは単に、目次を出力させたい場所で、

目次の出力

```
\tableofcontents
```

と命令すればいいのです。非常に簡単ですね。ただし、目次のページが出来上がるためには、一般には **platex** を 2 回実行する必要があります。最初のコンパイルで目次情報を集めソースを自動作成しておき、2 度目で実際に DVI ファイルに変換するのです。

目次の例

1	では ”こんにちはラテフ!” から	2
1.1	ソースを emacs で編集	2
1.2	platex で DVI ファイルへ	2
1.3	xdvi によりレイアウト確認	3
1.4	まとめ	4
1.4.1	L ^A T _E X で整形された文書を作成する手順	4
1.4.2	最小限度記述が必要なコマンド 3 行	4
2	章・節建て	6
3	タイトルと著者	8
4	目次	10
5	ヘッダやフッタ	11
5.1	形式	11
5.2	ページ番号	11
6	数式	13
6.1	数式モード	13
6.1.1	eqnarray 環境	13
6.2	数式フォントと空白の扱い	14
6.2.1	英数字と空白	14
6.2.2	立体文字	15
6.2.3	ギリシャ文字	15

他にも `\listoffigures` や `\listoftables` により、図や表の一覧が生成されます。

5 ヘッダやフッタ

各ページの上部欄外がヘッダ、下部欄外がフッタです。ここにページ番号や日付やセクション番号とその表題等を載せます。少なくとも、ページ番号はヘッダかフッタのどちらかに必要でしょう。

5.1 形式

ヘッダフッタの形式を指定するコマンドがあります。

```
\pagestyle{スタイル名} \thispagestyle{スタイル名}
```

です。`\pagestyle` はそれ以降の全てのページに有効ですが、`\thispagestyle` はそのページのみ (臨時) に有効です。また、スタイル名は次のいずれかを選択します。既定は `plain` です。

表 1 ヘッダフッタスタイル

スタイル名	出力内容
<code>plain</code>	ヘッダは空。フッタにはページ番号が中央揃えで出力されます。
<code>empty</code>	ヘッダもフッタも空。
<code>headings</code>	ヘッダには章節の番号とその表題、およびページ番号が出力されます。フッタは空。
<code>myheadings</code>	ヘッダを自分で指定できますが、後で紹介する <code>fancyheadings.sty</code> を使う方がいいので、ここでは説明を省略します。

5.2 ページ番号

目次と本文のページ番号を別にしたい場合には、目次を出力した後、ページ番号を自分で指定する必要があります。ページ番号は `page` というカウンターに保持されていますから、その値を 1 に戻せばいいのです。

```
\setcounter{page}{1}
```

で旨く行きます。目次にもページ番号がつきますが、本文と同じでは見苦しいので一般にはローマ数字で出力します。数字のフォントを変更するコマンドは

```
\pagenumbering{フォント名}
```

フォント名は次表のいずれかです。既定は `arabic` です。

表 2 ページ番号のフォント指定

フォント名	出力内容
<code>arabic</code>	アラビア数字 1, 2, 3, 4, ...
<code>roman</code>	小ローマ数字 i, ii, iii, iv, ...
<code>Roman</code>	大ローマ数字 I, II, III, IV, ...
<code>alph</code>	英小文字 a, b, c, d, ...
<code>Alph</code>	英大文字 A, B, C, D, ...

練習 6

目次と本文に別のページ番号をつけてみましょう。次のような内容で `ex6.tex` を編集して、レイアウトを確かめましょう。

```

\documentclass[a4j]{jarticle}
\title{超電導薄膜の作成技術：最近の展望}
\author{松田彩}
\date{平成 8 年 10 月 22 日}

\begin{document}
\maketitle
\thispagestyle{empty}

\newpage
\pagenumbering{roman}
\tableofcontents

\newpage
\pagenumbering{arabic}

\section{はじめに}
最近、超電導薄膜の作成技術の進歩に応じて、その産業における利用への期待は
急速に高まっています。...
\subsection{作成方法の発展状況}
\subsection{酸素含有量の影響}
\subsection{磁束の封入}


\newpage
\section{可塑性基板との組み合わせ}

\newpage
\section{アニーリングについて}

\newpage
\section{元素別に有望な組成比}
\subsection{Bi 系}
\subsection{Y 系}

\newpage
\section{今後の課題}
\end{document}

```

 `\newpage` は改頁コマンドです。それらしい骨格ができあがりませんか？ タイトルや著者のフォントをいじってみましょう。新しい内容を目次に反映させるには、`platex` を 2 回実行してください。

6 数式

数式は L^AT_EX 2_ε の最も得意とするところです。これだけの印字品質は他のワープロでは得られません。

6.1 数式モード

数式を書くには、数式モードに移行する必要があります。モードは、テキストに混ぜるためのインラインモードと、改行して数式だけの段落を形成するディスプレイモードがあり、開始と終了の記号 (あるいはコマンド) で括ります。実際の印字結果に差がありますが表記自身は共通です。

表 3 数式モードの種類

名前	開始と終了の記号	例
math	<code>\begin{math}...\end{math}</code> または <code>\$...\$</code> または <code>\(...\)</code> で括る。	テキストに $a + b = z$ のように混ざります。積分記号 $\int_a^b f(x)dx$ や和記号 $\sum_{k=1}^n a_k x^k$ や分数 $\frac{x}{1+x^2}$ が行間間隔を乱さないように狭く表示されます。
displaymath	<code>\begin{displaymath}...\end{displaymath}</code> または <code>\$\$...\$\$</code> または <code>\[...\]</code> で括る。	テキスト部分とは独立の 1 行 $\int_a^b f(x)dx$ を作ります。
equation	<code>\begin{equation}...\end{equation}</code> で括る。 <code>displaymath</code> と同じく 1 行のみ数式を表示します。ただし、番号が自動的に割り当てられます。	テキスト部分とは独立の 1 行 $\int_a^b f(x)dx \quad (1)$ を作ります。通し番号も付きます。
eqnarray	<code>\begin{eqnarray}...\end{eqnarray}</code> で括る。複数行の数式を書く場合に使います。基本的には 1 行毎に番号が付けられます。	段落を改めて複数行の数式を描きます。 $F(x) = \int_{-\infty}^x f(\xi)d\xi \quad (2)$ $= \log \tan(x) + 1 \quad (3)$ 全ての行に番号がつけられます。
eqnarray*	<code>\begin{eqnarray*}...\end{eqnarray*}</code> で括る。複数行の数式を書く場合に使います。番号が付きません。	段落を改めて複数行の数式を描きます。 $F(x) = \int_{-\infty}^x f(\xi)d\xi$ $= \log \tan(x) + 1 $ 番号が付きません。

6.1.1 eqnarray 環境

この環境では、数式を行列のように複数行書く事が可能ですが、後に説明する表と同様の書式を使いません。それは

表や行列を書く上での共通書式

行の終了記号に `\\`、各欄の区切り記号に `&` を用いる

です。具体的な例を挙げますから、ソースと出力を見比べてください。

```

\begin{eqnarray}
G(x) & = & (n+m)(c+d) \\
& < & (n+m)(c+d+1)
\end{eqnarray}

```

$$G(x) = (n+m)(c+d) \quad (4)$$

$$< (n+m)(c+d+1) \quad (5)$$

6.2 数式フォントと空白の扱い

では単純な数式を入力して出力を見ましょう。

例題 7

次のような内容で samp7.tex を編集して、出力をみましょう。

```

\documentclass[a4j]{jarticle}
\begin{document}

ニュートンの運動方程式は、力を  $f$ 、質量を  $m$ 、加速度を  $a$  とし
て、
\[
f = ma
\]
と表現される。

放射線には  $\alpha$  線、 $\beta$  線、 $\gamma$  線がある。

\end{document}

```

出力結果

ニュートンの運動方程式は、力を f 、質量を m 、加速度を a として、

$$f = ma$$

と表現される。

放射線には α 線、 β 線、 γ 線がある。

練習 7

displaymath 環境： \displaystyle を $\mathbb{...}$ にしてみたり、equation 環境を使ってみたりしましょう。また、

```

ニュートンの運動方程式は、力を  $f$ 、質量を  $m$ 、加速度を  $a$  として、

$$f = ma$$

と表現される。

```

としても出力結果は同じはずですが、読み易さを考えて、例題のように書くのがいいでしょう。

6.2.1 英数字と空白

英数字には、普通の *italic* よりも幅広な *mathitalic* というフォントが使われます。また、通常のテキストと異なり、空白文字は無視されます。従って

```

\[
a+b=c a + b = c
a+b
=c
\]

```


はいずれも $a + b = c$ となります。

6.2.2 立体文字

三角関数等は変数と区別するために、立体文字を使います。したがって、ただ単に $\$sin x\$$ と表記すると $sinx$ になってしまうので、 $\$sin x\$$ と表記して $sin x$ を得ます。このような関数名や記号を L^AT_EX 2_ε では Log 型関数と呼びます。

表 4 立体文字で出力される関数名や記号 : Log 型関数

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

出力は省略しています

6.2.3 ギリシャ文字

ギリシャ文字は以下の表記を用います。ギリシャ文字の読み方が覚えられますね。

表 5 ギリシャ文字

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

練習 8

以下の各項のような出力結果を得るソースファイルを作成しなさい。

$$(a + b)(c + d) = ac + ad + bc + bd, \quad \exp[x + y] = \exp[x] \exp[y]$$
$$5[\Omega], \quad \sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta$$

練習 9

以下の数式ソースで間違っている箇所を指摘しなさい (3 箇所あります)。

```
\documentclass[a4j]{jarticle}
\begin{document}
\begin{equation}
\log(xy) = \log x + \logy
\end{equation}
```

6.3 数式記法

少し慣れたところで、数式記法を列挙しましょう。

6.3.1 上付き・下付き文字

上付きは $\hat{}$ (ハット)、下付きは $\underset{}{}$ (アンダースコア) の後の 1 文字もしくは中括弧 { } で囲まれた文字列が作用を受けます。

表記	実際の表示
<code>x^2</code>	x^2
<code>e^{-at}</code>	e^{-at}
<code>a_1, \dots, a_{12}</code>	a_1, \dots, a_{12}
<code>c_{\alpha}^{\gamma}</code>	c_{α}^{γ}
<code>e^{-\beta x^2}</code>	$e^{-\beta x^2}$
<code>kanji^{\text{漢字も}}_{\text{大丈夫}}</code>	$kanji^{\text{漢字も}}_{\text{大丈夫}}$

6.3.2 積分・級数和

`\int_{\text{下限}}^{\text{上限}}`, `\sum_{\text{初期値}}^{\text{終条件}}` のように表記します。

表記	実際の表示
<code>\int_0^x f(x')dx'</code>	$\int_0^x f(x')dx'$
<code>\sum_{k=1}^{\infty} a_k x^k</code>	$\sum_{k=1}^{\infty} a_k x^k$
<code>\oint_C f(s)ds</code>	$\oint_C f(s)ds$

6.3.3 根号

数 x の n 乗根は `\sqrt[n]{x}` と表記します。[n] を省略すると表示が無くなりますから、即ち平方根となります。

表記	実際の表示
<code>\sqrt{2}</code> , <code>\sqrt[n]{r}</code>	$\sqrt{2}$, $\sqrt[n]{r}$
<code> r =\sqrt{x^2+y^2+z^2}</code>	$ r = \sqrt{x^2 + y^2 + z^2}$
<code>\sqrt{\text{漢字も}}{\text{大丈夫}}</code>	$\sqrt{\text{漢字も}}{\text{大丈夫}}$

6.3.4 分数

`\frac{分子}{分母}` と表記します。複雑な式があると文字が小さくなるので、普通の大きさの文字を使うのには `\frac{\displaystyle 分子}{分母}`(これは分子が小さい文字になっている場合の修正) の様に `\displaystyle` を指定します。

表記	実際の表示
<code>\frac{1}{2} \frac{y}{x}</code>	$\frac{1}{2} \frac{y}{x}$
<code>\frac{x+y}{x-y} \frac{z}{1+z^2}</code>	$\frac{x+y}{x-y} \frac{z}{1+z^2}$
<code>\frac{y}{1 + \displaystyle \frac{1}{1+x}}</code>	$1 + \frac{y}{1+x}$
<code>\frac{漢字も}{大丈夫}</code>	漢字も 大丈夫

6.3.5 行列

行列は `array` 環境を使います。行内の要素を `&` で区切り `\\` で行を終了します (これは `eqnarray` 等と同じ)。

```
\begin{array}{ccc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}
```

$$\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{matrix}$$

行列の行数に合わせた大きさの括弧を表示する場合は `\left(` と `\right)` で `array` 環境を囲みます。

```
\left(
\begin{array}{c|cc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}
\right)
```

$$\left(\begin{array}{c|cc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \right)$$

少し文法めいた部分が長くなってしまいました。練習問題をして、気分を晴らしましょう？


練習 10

次のような出力を得るソースを考えなさい。

温度 T にある理想気体の Maxwell-Boltzmann 速度分布は、分子間相互作用ポテンシャルエネルギーがないものとしているので、分子の全エネルギー ϵ が運動エネルギー $\frac{mv^2}{2}$ に等しいことに注意して

$$f(v_x, v_y, v_z) dv_x dv_y dv_z = \left(\frac{m}{2\pi kT} \right)^{\frac{3}{2}} e^{-\frac{\epsilon}{kT}} dv_x dv_y dv_z$$

と書かれる。

 分数を囲む括弧は大きさを自動判別する `\left(` と `\right)` を用います。

練習 11

次のような出力を得るソースを考えなさい。


エネルギーギャップ E_G を持つ真性半導体の伝導電子密度を n 、空孔密度を p とおく。伝導電子と空孔をそれぞれ実効質量 m_e 、 m_h をもつ自由粒子とみなせば、

$$n = p = 2 \left\{ \frac{2\pi\sqrt{m_e m_h} kT}{h^2} \right\}^{\frac{3}{2}} e^{-E_G/2kT}$$

となること、またこの電子系の Fermi ポテンシャルは

$$\mu = \frac{1}{2}E_G + \frac{3}{4}kT \log \frac{m_h}{m_e}$$

であることが示される。ただし、 $E_G \gg kT$ とする。

 分数を囲む中括弧は大きさを自動判別する `\left\{` と `\right\}` を用いましょう。『十分大きい』の記号 \gg は `\gg` を使います。

練習 12

∂ (ラウンドディー)の表記は `\partial`、 \hbar の表記は `\hbar` であることを用いて、(1次元)1粒子の Schrödinger 方程式


$$\mathcal{H}\psi(x) = \left\{ -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right\} \psi(x) = E\psi(x)$$

を出力してみなさい。ハミルトニアンには筆記体 `\mathcal{H}` がそれらしい書体です。

練習 13

以下の積分公式を出力してみなさい。

$$\begin{aligned} \text{区間 } [a, b] \text{ における曲線 } y = f(x) \text{ の長さ} &= \int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \\ \int_0^1 (1-x^2) dx &= \frac{(2n)!!}{(2n+1)!!} \quad \int_0^{\frac{\pi}{2}} \sin^\alpha x dx = \frac{\sqrt{\pi}}{2} \Gamma\left(\frac{\alpha+1}{2}\right) / \Gamma\left(\frac{\alpha+2}{2}\right) \end{aligned}$$

 大ギリシャ文字を斜体にするには(フォントの指定)、`\mit` を使って `\mit\Gamma` とします。大きいスラッシュは `\Big/` を使ってください。

練習 14

以下の行列演算式を出力しなさい。

$$\text{座標の回転: } R(\theta) \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

7 図の貼り込み

TeXでも図形を描く機能はありますが、簡単なものしかありませんから、せいぜいフローチャート程度が関の山でしょう。図(画像ではありませんから、念のため)はそれ専用のアプリケーションで作成して、TeXの文章に貼り込みましょう。

7.1 図を入れる空白を用意しておき、後に糊で貼る

電子的に図を貼る方法の前に、原始的ですがどんな図や写真でも貼り付けられる方法を説明しましょう。見出しの通り、 \LaTeX には場所だけを確保させるのです。場所の確保には`\hspace{長さ}`を用います。また、前後の段落との間隔の自動調整のために`figure`環境を使います。

`\begin{figure}...\end{figure}`で囲むと、前後の段落とに適切な間隔を空けてレイアウトされます。`[htbp]`は図の配置指定です。`h`(here)はコマンドの置かれた直後、`t`(top)はコマンドの置かれた頁の上部、`b`(bottom)は下部、`p`(post?)は最後にまとめて出力するという意味です。左側の指定から順にレイアウト可能か \LaTeX が判断して、可能ならばそのように出力します。`[h]`のみを指定すると、`h`が失敗した場合、保留となって以降全ての図表は最後に出力されます。するとメモリ不足に陥る危険性がありますから、必ず`[htbp]`や`[hbtbp]`等と指定しておきましょう。


また、この`figure`環境内では、`\caption{...}`により、自動的に番号をつけた図のキャプションが書けます。後で出て来る`table`環境も同様に、前後の段落との空白調整とキャプションへの自動的な番号付けがされます。

例題 8

図を糊で貼る場所を用意しましょう。次の内容で`samp8.tex`を作成して`platex`でコンパイルし`xdvi`で空白の空き具合を確かめましょう。

```
\documentclass[a4j]{jarticle}
\def\いろは{\いろはにほへとちりぬるをわかよたれそつねならむういのおくやま
きょうこえてあさきゆめみしいねもせず。}
\begin{document}
\いろは\いろは
\begin{figure}[htbp]
\vspace{4cm}
\caption{後で糊付けします}
\end{figure}

\いろは
\end{document}
```

 このような図表のひとかたまりをフロートと呼びます。 \LaTeX は、レイアウトを自動調整する際、テキストとフロートが1頁に占める割り合いを調整します。その結果、自分の指定した場所に図が入らない(即ち、 \LaTeX が好ましくないレイアウトと判断してしまう)場合が起こります。この問題の回避方法は中級コースで述べます。今は、 \LaTeX に怒られない図や表の配置を考えましょう。

7.2 EPS形式の図を貼り込む

図の保存形式は多種多様ですが、拡大縮小が自由なPostScript形式がUNIX系では標準的な地位を占めています。PostScript(以降PSと略す)も機能拡張が進んでいるので、様々なアプリケーションで共通に使

えるものに制限 (Encapsulated) した PS を $\text{\LaTeX} 2_{\epsilon}$ の文章に貼り込むためのパッケージが `graphicx` です。 \LaTeX が行うことは、EPS ファイルの大きさを判断して空白を確保するだけです。実際に図を BitMap に変換するのは外部プログラム (この場合は GhostScript) に任せます。

このパッケージを利用するには、まず、`\documentclass` の次の行で `\usepackage` で `graphicx` をオプション指定しておきます。

```
\usepackage{graphicx}
```

そして、図をいれたい場所で

```
\includegraphics[size]{filename}
```

と命令します。

例題 9

`graphicx` を利用して `\includegraphics` コマンドで図を貼り込んでみましょう。

```
\documentclass[a4j]{jarticle}
\usepackage{graphicx}
\begin{document}
\いろは\いろは
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=5cm]{resurge1.eps}
\caption{リサージュ図形}
\end{center}
\end{figure}

\いろは
\end{document}
```

graphicx で EPS 図形を貼り込んだ例

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

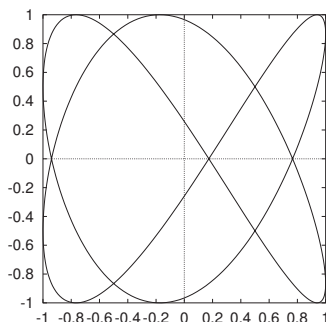


図 1 リサージュ図形

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

8 表

表は罫線が必要なければ `tabbing` で作成します。この環境では、単なる基本的な文字の整列機能だけが用意されています。ところで、特に日本人は『表』という罫線を使いたがる傾向にあるようです。罫線はテキストで実現するのは難しいので、全体として処理は複雑になってしまいます。L^AT_EX では、あまり複雑な罫線は引けませんので、ビジネス用ワープロに比べて不満が残るかもしれません。が、学術論文では、罫線は実線のみしかも線の太さも 1 種類である場合がほとんどです。したがって、L^AT_EX で十分です。フォントも種類の使いすぎが見苦しいのと同様に罫線も乱用は控えましょう。

8.1 tabular 環境

さて、罫線が引ける表作成のための環境は `tabular` です。書式の細かい指定は後回しにして、取り敢えず表を作成してみましょう。

例題 10

```
\documentclass[a4j]{jarticle}
\begin{document}
\いろは

\begin{tabular}[t]{l|c|r|p{4cm}|}
\hline
試料番号 & 圧力 [Pa] & 温度 [K] & 表面形態 \\ \hline
1 & 10 & 300 & 黒色。ガサガサ。 \\
2 & 10 & 400 & 金属光沢。滑らか。 \\
3 & 20 & 300 & \\
4 & 20 & 400 & 金属光沢。温度上昇のためか、粒界が見える。 \\ \hline
\end{tabular}

\いろは
\end{document}
```

出力例 : tabular

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

試料番号	圧力 [Pa]	温度 [K]	表面形態
1	10	300	黒色。ガサガサ。
2	10	400	金属光沢。滑らか。
3	20	300	
4	20	400	金属光沢。温度上昇のためか、粒界が見える。

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

これだけでは、前後の段落との間隔が変ですね。この点は、図と同様に `table` 環境で囲むことで解決できますから、後で練習しましょう。

では書式を簡単に説明します。まず全体は

```
\begin{tabular}[t]{列の出力形式}
  項目 (1,1) & 項目 (1,2) & ... \\
  項目 (2,1) & ... \\
\end{tabular}
```

のような感じです。列の出力形式は (行毎に繰り返されます)、テキストの揃え方に関しては

テキストの位置を揃えるが折り返しができない: l, c, r
幅を指定してテキストの折り返しを認める: p{列幅}

と設定できます。また、縦罫線を引く場合には " | " 或は " || (2重線)" を使います。また、数式のところで説明しましたが、表においても

表や行列を書く上での共通書式

行の終了記号に \\, 各欄 (項目) の区切り記号に & を用いる

を覚えましょう。横罫線は \\ の後に \hline を使います。\\hline を 2 つ書けば 2 重の横罫線になります。

練習 15

表と前後の段落との見栄えのよいレイアウトを行うには、**table** 環境を用いてフロートにします。figure 環境とほとんど同じですが、表のキャプションが "表 3.." のように始まり、番号が別に管理される点が異なります。そこで、例題の表を table 環境で囲って出力してみましょう。

table 環境による表の出力例

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

表 6 試料作成時のスパッタリング条件

試料番号	圧力 [Pa]	温度 [K]	表面形態
1	10	300	黒色。ガサガサ。
2	10	400	金属光沢。滑らか。
3	20	300	
4	20	400	金属光沢。温度上昇のためか、粒界が見える。

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

込み入った表を書くには、もう少しコマンドを覚える必要があります。もっとも良く使うのが、**multi-column** と **cline** です。この 2 つを使うと、カラムのレイアウトがかなり自由にできます。

8.1.1 cline

特定のカラムを指定してそこに罫線をひきます。命令できる場所は、\hline と同じく \\ の後です。

```
\cline{開始カラム-終了カラム}
```

の様に書きます。複数指定することができます。

8.1.2 multicolumn

複数カラムを1つのカラムにまとめて、出力形式を変更することができます。

```
\multicolumn{カラム数}{出力形式}{テキスト}
```

という書式を用います。

例題 11

`\cline` や `\multicolumn` を使って少し複雑な表を作成しましょう。

```
\documentclass[a4j]{jarticle}
\begin{document}
\begin{table}[htbp]
\centering
\begin{tabular}[t]{|l|c|r|p{4cm}|}
\hline
試料番号 & \multicolumn{2}{c|}{スパッタ条件}
& \hfill{}表面形態\hfill{} \\ \cline{2-3}
& 圧力 [Pa] & 温度 [K] & \\ \hline
1 & 10 & 300 & 黒色。ガサガサ。 \\ \hline
2 & 10 & 400 & 金属光沢。滑らか。 \\ \hline
3 & 20 & 300 & \\ \hline
4 & 20 & 400 & 金属光沢。温度上昇のためか、粒界が見える。 \\ \hline
\end{tabular}
\end{table}
\end{document}
```


 `\multicolumn` を使ったときに、出力形式の縦罫線の指定をつい忘れてしまうことが多いので注意しましょう。すなわち、`{c|}` とすべきところを `{c}` とつい書いてしまうのです (最初に全体を指示していると思っ込んでいるせいでしょう)。

表 : `cline,multicolumn` 命令を使用

試料番号	スパッタ条件		表面形態
	圧力 [Pa]	温度 [K]	
1	10	300	黒色。ガサガサ。
2	10	400	金属光沢。滑らか。
3	20	300	
4	20	400	金属光沢。温度上昇のためか、粒界が見える。

9 フォント

英数文字はかなり自由に選べますが、日本語書体は基本的には2種類 (一般には明朝とゴシック) しか使えません。また、幾つかの文字は制御文字として予約されてしまい、その文字自身を表示させるための命令が必要となります。

9.1 特殊文字

バックスラッシュ文字 '\ ' が先頭についた文字列はコマンドと解釈されてしまいます。したがって、文字として '\ ' を表示する場合は他の表記をしなければなりません。このように、 \LaTeX では全ての制御コマンドを可視文字列で表記しているため、幾つかの文字は "特殊文字" と解釈されます。したがって、文字として表示したい場合は特別な表記が必要です。

表 7 特殊文字とその表記

表示	表記	表示	表記	表示	表記
\$	\\$	%	\%	~	\symbol{'176}
%	\%	}	\}	^	\symbol{'136}
-	_	{	\{	\	{\tt\symbol{'134}}
#	\#				

シャープ記号 '#' 等はその文字の前にバックスラッシュ '\ ' を付けるという規則に従います (一般にはエスケープと呼ばれます)。しかし、チルダ '~'、ハット '^'、およびバックスラッシュ '\ ' をこの規則で表すと、他の命令になってしまいます。\\ は強制改行のコマンドですし、\~ と \^ はアクセント記号をつけるコマンドとして既に定義されてしまっているのです。

9.2 英数字の書体

前節の特殊文字を含めて、標準で使える英語の書体を下の表に掲げます。バックスラッシュをわざと \symbol{'134} で定義してあります。この表を見ると \tt\symbol{'134} でなければならない理由が理解できます。

表 8 英数文字記号の書体

書体 (命令)	表示
ローマン (\rm)	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNopqrstuvwxyz 1234567890!@*()-+= '[] : ; " ' , . / ? < > \$ % _ # % } { ^ ~
イタリック (\it)	<i>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNopqrstuvwxyz</i> <i>1234567890!@*()-+= '[] : ; " ' , . / ? < > \$ % _ # % } { ^ ~</i>
タイプライター (\tt)	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNopqrstuvwxyz 1234567890!@*()-+= '[] : ; " ' , . / ? < > \$ % _ # % } { ^ ~
斜体 (\sl)	<i>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNopqrstuvwxyz</i> <i>1234567890!@*()-+= '[] : ; " ' , . / ? < > \$ % _ # % } { ^ ~</i>
サンセリフ (\sf)	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNopqrstuvwxyz 1234567890!@*()-+= '[] : ; " ' , . / ? < > \$ % _ # % } { ^ ~
ボールド (\bf)	abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNopqrstuvwxyz 1234567890!@*()-+= '[] : ; " ' , . / ? < > \$ % _ # % } { ^ ~
スモールキャピタル (\sc)	ABCDEFgHIJKLmNOPQRStUVWxyz ABCDEFgHIJKLmNOPQRStUVWxyz 1234567890!@*()-+= '[] : ; " ' , . / ? < > \$ % _ # % } { ^ ~

9.3 大きさの指定

地の文の大きさに対して相対的な比率を指定するコマンドを用いて大きさを変えます。また、フォントの変更と同時に指定できます。(L^AT_EX 2.09 では大きさの指定をフォントの指定の前に置く必要がありました。L^AT_EX 2_ε では順序を気にする必要はありません。)

表 9 フォントの大きさを指定する命令

命令	表示例
<code>\tiny</code>	size サイズ 大中小
<code>\scriptsize</code>	size サイズ 大中小
<code>\footnotesize</code>	size サイズ 大中小
<code>\small</code>	size サイズ 大中小
<code>\normalsize</code>	size サイズ 大中小
<code>\large</code>	size サイズ 大中小
<code>\Large</code>	size サイズ 大中小
<code>\LARGE</code>	size サイズ 大中小
<code>\huge</code>	size サイズ 大中小
<code>\Huge</code>	size サイズ 大中小

例題 12

フォントの大きさ(や書体)を変えてみましょう。次のソースを基にしていろいろ楽しんでみましょう。

```
{\large \bf 修士の皆さんへ}

\hfill{平成 8 年 3 月 11 日}

\begin{center}
\Large\bf \LaTeX{} 春休み短期講習会開催のお知らせ
\end{center}

理系論文に最適の\LaTeX{}を覚えたいという貴方に良い知らせです。...
\いろは\いろは\いろは

\begin{flushright}\obeylines\large
物質工学科薄膜物性研究室
\it 5308 号室, 内線: 2978
\end{flushright}
```

10 箇条書き (List)

LaTeX では箇条書のための環境として、`itemize`、`enumerate`、`description` が用意されています。

10.1 マーク付き箇条書き : `itemize`

`itemize` は普通の箇条書きで、項の始まる印が先頭に付きます。書式は簡単でして、以下の通りです。

全体を `\begin{itemize}` と `\end{itemize}` で囲みます。 `\item` を置いて各条項を書きます。

例題 13

`itemize` 環境を使ってみましょう。これからは、最低限必要な **3 行** は特に明記しませんから、下のようなソースに必ず自分で補ってください。

```
\いろは
\begin{itemize}
\item 第 1 箇条を書きましょう。
\item 第 2 箇条を書きましょう。第 2 箇条を書きましょう。折り返しや 2 行目の字
下げの具合に注意してください。item 間では空白が生じ、日本語ワープロ (原稿
用紙に書くように、行間が一定) ばかりを見てきたので、違和感があるかも知れ
ません。
\item 第 3 箇条の中で itemize を入れ子にして使います。
\begin{itemize}
\item 第 3 箇条の 1
\item 先頭の印が変化しています。
\end{itemize}
\end{itemize}
\いろは
```

箇条書: `itemize`


いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

- 第 1 箇条を書きましょう。
- 第 2 箇条を書きましょう。第 2 箇条を書きましょう。折り返しや 2 行目の字下げの具合に注意してください。item 間では空白が生じ、日本語ワープロ (原稿用紙に書くように、行間が一定) ばかりを見てきたので、違和感があるかも知れません。
- 第 3 箇条の中で `itemize` を入れ子にして使います。
 - 第 3 箇条の 1
 - 先頭の印が変かしています。

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

練習 16

`itemize` は 4 段階の深さまで定義されています。項の始まりの印は、第 1 の深さでは ●、第 2 では – でした。では、第 3 と第 4 ではどのような印が設定されているでしょうか、調べてください。

 `\begin{itemize}... \end{itemize}` を 4 回入れ子に書いて調べましたか? ご苦労様でした。実はそれぞれの印は `\labelitemi`, `\labelitemii`, `\labelitemiii`, `\labelitemiiii` に定義されています。従って、単に

```
\labelitemi, \labelitemii, \labelitemiii, \labelitemiv
```

と書くだけで印が表示されます。正解は、`'•'`, `'-'`, `'*'`, `'.'`

10.2 番号付き箇条書き : enumerate

項の始まりに、序列を示す数字やアルファベットが印として付きます。書式は `itemize` と変わりありません。

例題 14

番号付きの箇条書 `enumerate` を使ってみましょう (3 行を補うのを忘れないように)。

```
\いろは
\begin{enumerate}
\item イチゴ
\item 人参
\begin{enumerate}
\item 人参サラダ
\item スイートキャロット
\end{enumerate}
\end{enumerate}
\いろは
```

箇条書: enumerate

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

1. イチゴとイチコと今川焼きと石狩鍋と石焼き芋とインドカレーと猪の肉はボタンと石川県は日本海に面しているので魚が旨いと...
2. 人参
 - (a) 人参サラダ
 - (b) スイートキャロット
 - i. 第 3 レベル
 - A. 第 4 レベル

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

ここでも深さは 4 段階まで定義されています。それぞれの印は

```
\labelenumi, \labelenumii, \labelenumiii, \labelenumvi
```

を見ればいいのです。既定では 数字, 括弧付きの英小文字, 小ローマ数字, 英大文字となっています。

練習 17

itemize では、印を変えたいとは思わないでしょうが、enumerate で付けられる数字の種類は変えたいと思うことでしょう。理由は中級コースで理解してもらうことにして、ここでは方法のみを練習しましょう。例えば、第1段階の数字を四角括弧付きのアラビア数字に設定するには

```
\def\labelenumi{[\arabic{enumi}]} または
\renewcommand\labelenumi{[\arabic{enumi}]}
```

とします。page 番号の場合と同様の種類が使えます (p10 表2 参照)。

10.3 見出し語付き箇条書き : description

辞書のように用語を見出し (太字) にして、説明文を続けて書いていきます。itemize や enumerate と少し書式が異なり、

```
\item[見出し]....
```

と記述します。私はどうもこの体裁は好きになれません。

例題 15

見出し語付きの箇条書きの例です。

```
\いろは
\begin{description}
\item[見出し語]
説明文.....
.....
\item[長い見出しを書く和不揃い]
説明文.....
.....
\end{description}
\いろは
```

箇条書き : description

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

見出し語 説明文.....
.....

長い見出しを書く和不揃い 説明文.....
.....

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

一般の List 環境について

箇条書きは共通の雛形 list のパラメータを設定して得られています。したがって、中級以上になれば自分でもあるリスト環境を定義できるようになります。それは後のお楽しみとして、リストで定義されている他の環境を紹介します。

10.4 引用環境 : quote, quotation

理工系の論文ではあまり使われませんが、文系の論文では他の著者の文章をそのまま引用するときには、左 (右) を字下げしたりして明確に区別しなければなりません。そのための環境が `quote` と `quotation` です。


2つの環境の違いは段落の示し方にあります。すなわち、`quote` では段落の先頭で字下げをせずに、少し間隔を空けます。他方、`quotation` では段落の先頭を字下げして、行間隔は一定のままです。

例題 16

文章引用のための環境 `quote` と `quotation` の違いを確かめましょう。

```
quote 環境では
\begin{quote}
いろは\parいろは
\end{quote}
```

```
quotation 環境では
\begin{quotation}
いろは\parいろは
\end{quotation}
```

 `\par` は段落を終了させるコマンドです。紙面の節約のために (本当のところは空行が多いと間が抜けている感じがするので、体裁をよくするために)、空行の代わりに用いました。ただし、段落が判別しにくくなりますから、どうしても必要な場合以外には使わない方がいいでしょう。

quote と quotation の違い

quote 環境では


```
いろはにほへとちりぬるをわかよたれ
そつねならむういのおくやまきようこ
えてあさきゆめみしいねもせず。

いろはにほへとちりぬるをわかよたれ
そつねならむういのおくやまきようこ
えてあさきゆめみしいねもせず。
```

quotation 環境では

```
いろはにほへとちりぬるをわかよた
れそつねならむういのおくやまきよう
こえてあさきゆめみしいねもせず。

いろはにほへとちりぬるをわかよた
れそつねならむういのおくやまきよう
こえてあさきゆめみしいねもせず。
```

 `quote` と `quotation` の行間の違いを明確にするために、後で説明する `minipage` 環境を使って左右に並べました。例題のソースではこのような出力になりません。

10.5 文字揃え環境 : flushleft, center, flushright

リストのついでに文章の左揃え、中央揃え、右揃えの環境を説明します。中央揃えはいろいろな場面で必要になりますが、左揃えや右揃えはタイトルの作成や手紙を書く場合を除いてあまり必要ないでしょう。使い方は簡単で次の例題で十分理解できます。

例題 17

文字揃えの環境を使ってみましょう。

```
\begin{flushleft}
古池や蛙飛び込む水の音\\
山路来てなにやらゆかしすみれ草
\end{flushleft}
```

```
\begin{center}
古池や蛙飛び込む水の音\\
山路来てなにやらゆかしすみれ草
\end{center}
```


```
\begin{flushright}
古池や蛙飛び込む水の音\\
山路来てなにやらゆかしすみれ草
\end{flushright}
```

出力例：左揃え、中央揃え、右揃え

古池や蛙飛び込む水の音
山路来てなにやらゆかしすみれ草

古池や蛙飛び込む水の音
山路来てなにやらゆかしすみれ草

古池や蛙飛び込む水の音
山路来てなにやらゆかしすみれ草

 \\ は段落内部の強制改行です。改行のたびにこの命令を入力していくのはかなり面倒です。その場合には `\obeylines` を使うと、強制改行コマンド `\\` を一々書かなくとも済みます。すなわち、次のように書くと上の例と同じ出力が得られます。

obeylines の使い方の例

```
\begin{flushleft}\obeylines
古池や蛙飛び込む水の音
山路来てなにやらゆかしすみれ草
\end{flushleft}
```

```
\begin{center}\obeylines
古池や蛙飛び込む水の音
山路来てなにやらゆかしすみれ草
\end{center}
```

```
\begin{flushright}\obeylines
古池や蛙飛び込む水の音
山路来てなにやらゆかしすみれ草
\end{flushright}
```

10.6 ミニページ : minipage

通常のページの中に、(小さな)幅を指定してページを構築する環境が **minipage** です。とても重宝するので、実際この教材中でも随所に登場します。書式は

```
\begin{minipage}[位置]{幅} ... \end{minipage}
```

となっています。'位置'は周りの行との垂直方向の関係調節で、t, c, b を指定します (既定は c)。t では minipage の上部、c では中央、b では底が周りの行と揃えられます。

例題 18

非常に便利な minipage 環境になれましょう。ただし、この例題のように段落内に使うことは滅多になりません。行の基準線を引きましたから、'位置' オプションの感じを理解してください。

```
\rule[0mm]{1cm}{0.4pt}
\begin{minipage}[t]{0.25\textwidth}\いろは\end{minipage}
\rule[0mm]{1cm}{0.4pt}
\begin{minipage}[c]{2.5cm}\いろは\end{minipage}
\rule[0mm]{1cm}{0.4pt}
\begin{minipage}[b]{15zw}\いろは\end{minipage}
```

minipage 環境

<p>_____ いろはにほへとちりぬる _____ をわかよたれそつねならむ ういのおくやまきようこえ てあさきゆめみしいねもせず。</p>	<p>いろはにほへと ちりぬるをわか よたれそつねな らむういのおく やまきようこえ てあさきゆめみ しいねもせず。</p>	<p>いろはにほへとちりぬるをわかよ たれそつねならむういのおくやま きようこえてあさきゆめみしいね もせず。</p>
--	--	--

練習 18

minipage を使って、以下のように二つの小さな図面を横に並べるソースを考えてみなさい。

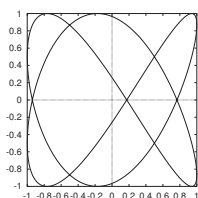


図 2 左側の図

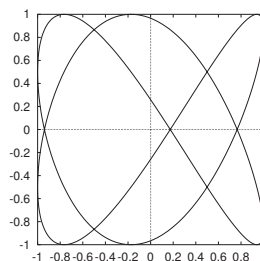



図 3 右側の図ちょっと大きい

11 空白制御

\LaTeX が自動的に文字列をレイアウトしてくれるので、書き手は内容にのみ集中できます。しかし、時々どうしてもレイアウトが気に入らない場合が生じます。そこで自分の希望のレイアウトを得るために、空白の制御方法を説明します。ただし以下の注意を心に留めておいてください。それは、初心者のうちには正しい用法を理解しないで、その場凌ぎに空白制御命令や強制改行命令 \backslash を使ってしまう傾向が見受けられますが、なるべく \LaTeX の意見を聞いてそのような対処療法ではなく正しい理解に基づく根本的な解決を心掛けるようにしましょう(自戒を込めて)。

 実は、空白制御は段落の先頭や段落内部、諸々の環境内部で幾つかの変数によって制御されています('sep' や'skip' を含むもの)。本来はそれらの変数の設定を変更すべきで、その場凌ぎに空白を変えるのは一言でいえば進歩がないやり方なのです。しかし、『文法を知らなくても英語が話せば用が足りる』と同様に『 \LaTeX の文法を知らなくても文章が書ければいいのだ、なにしろ道具なのだから』という姿勢はとても正しく、なにも全てを知る必要はないでしょう。

11.1 垂直方向の空白

11.1.1 空白を入れる命令

その場所にだけ臨時に空白を入れる命令を次の表にまとめます。表中の'長さ'は mm, cm, pt, em(文字 M), ex(文字 x) 等を単位に指定します。負の値も指定できて、その場合は詰めることとなります。

表 10 空白制御の方法

コマンド	機能
\backslash vspace{長さ}	垂直方向に'長さ'の空白を入れる。頁の最初等では無効となる。
\backslash vspace*{長さ}	垂直方向に'長さ'の空白を入れる。頁の最初等でも有効。
\backslash vfill	自動調節のための垂直方向に伸縮自在な空白。
\backslash [長さ]	改行後、次の行との間に'長さ'の空白を入れる。

例題 19

空白を入れて垂直方向の位置を微調する方法を練習しましょう。

```


\begin{minipage}[t]{1cm}ABC\par\vspace{-4mm}DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\par\vspace{-2mm}DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\par{}DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\par\vspace{2mm}DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\par\vspace{4mm}DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\par\vspace{6mm}DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\backslash[-4mm]DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\backslash[-2mm]DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\backslash\backslashDEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\backslash\backslash[2mm]DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\backslash\backslash[4mm]DEF\end{minipage}
\begin{minipage}[t]{1cm}ABC\backslash\backslash[6mm]DEF\end{minipage}

```

垂直方向の空白制御

```

ABC ABC ABC ABC ABC ABC DEF ABC ABC ABC ABC ABC
DEF DEF DEF DEF DEF DEF DEF DEF DEF DEF DEF DEF
  
```

 段落が変わる場合、強制改行\\による改行で入る空白 `baselineskip` の他に `parskip` が加わるので、段落改行`\par`の前後の行間隔が広がる筈です。ところが、`minipage` 環境では `parskip` が 0 になってしまうようで、例題のように同じ間隔が現れました。

11.1.2 行間隔を制御する命令

段落モードでの行と行の間隔は `\baselineskip` と `\baselinestretch` で制御します。

例題 20

`baselineskip` の値を変えてみましょう。

```

\noindent
\begin{minipage}[t]{0.32\textwidth}\parskip=10pt
baselineskip は標準 (=15pt) 設定。 \par\いろは
\end{minipage}\hfill
\begin{minipage}[t]{0.32\textwidth}\parskip=10pt\baselineskip=10pt
baselineskip を 10pt に設定。 \par\いろは
\end{minipage}\hfill
\begin{minipage}[t]{0.32\textwidth}\parskip=10pt\baselineskip=20pt
baselineskip を 20pt に設定。 \par\いろは
\end{minipage}
  
```

行間隔の制御 : `baselineskip`


`baselineskip` は標準 (=15pt) 設定。 `baselineskip` を 10pt に設定。

`baselineskip` を 20pt に設定。

いろはにほへとちりぬるをわかよ
たれそつねならむういのおくやま
きょうこえてあさきゆめみしいね
もせず。

いろはにほへとちりぬるをわかよ
たれそつねならむういのおくやま
きょうこえてあさきゆめみしいね
もせず。

いろはにほへとちりぬるをわかよ
たれそつねならむういのおくやま
きょうこえてあさきゆめみしいね
もせず。

 地の文の改行の様子に近づけるために、`minipage` 環境内部で `parskip` を 10pt に設定しました。従って、段落間同士の改行間隔が段落内の改行間隔よりも 10pt 広がっています。

11.2 水平方向の空白

文字間の調節のために固定幅の空白をいれる命令と、任意の指定幅を入れて命令とがあります。固定幅のものは数式の場合に重要です。表中の \sqcup は半角スペースを意味します (スペース自身を描けないので)。

表 11 水平方向の空白調節命令

ほぼ固定幅の空白：主として文字間調整	
<code>\quad</code>	そこで使われる文字ポイント幅の空白
<code>\qqquad</code>	<code>\quad</code> の 2 倍の空白
<code>\,</code>	<code>\quad</code> の 3/18 程の空白
[Ⓐ] <code>\:</code> (または <code>\></code>)	<code>\quad</code> の 4/18 程の空白
[Ⓐ] <code>\;</code>	<code>\quad</code> の 5/18 程の空白
[Ⓐ] <code>\!</code>	<code>\quad</code> の 3/18 程の負の空白
<code>\sqcup</code>	<code>\quad</code> の 6/18 程の空白。
<code>~</code>	<code>\sqcup</code> と同じ幅の空白。改行を抑制する働きがある。
<code>\/</code>	イタリック文字の後のごく僅かな空白
指定幅の空白調整	
<code>\hspace{幅}</code>	水平方向に'幅'の空白を入れる。
<code>\hspace*{幅}</code>	水平方向に'幅'の空白を入れる。行の最初でも有効。
<code>\hfill, \fill</code>	自動調整のための水平方向に一杯に伸びる空白。

[Ⓐ] は数式モードでのみ使えます。

例題 21

水平方向の指定幅空白命令 `hspace` を使ってみましょう。負の幅も指定できることを確認してください。また、幅が自動的に伸びる `hfill` を使って、中央揃えや右寄せを実現してみましょう。これをうまく使うと均等割りが可能となります。

```

\hspace*{20mm}
\hfill{}
\begin{minipage}[t]{50mm}
\noindent
ABC\hspace{-20mm}DEF\\
ABC\hspace{-10mm}DEF\\
ABC\hspace{0mm}DEF\\
ABC\hspace{10mm}DEF\\
ABC\hspace{20mm}DEF\\
\end{minipage}
\hfill{}
\begin{minipage}[t]{65mm}
愛\hfill{}夢\\
愛\hfill{}夢\hfill{}星\\
愛\hfill{}夢\hfill{}星\hfill{}花\\
\hfill{}終わらない\\
\hspace*{\fill}終わり\\
\mbox{~}\hfill{}中央\hfill{}~
\end{minipage}
\hfill{}

```


12 枠囲い

学会発表等では OHP やポスターを使いますが、効果的なプレゼンテーションを行うには、カラーでしかも凝ったデザインが必要でしょう。L^AT_EX はこのような事は苦手です。基本的にモノクロですし、大きなフォントも扱えませんから... しかし、ある程度の事はできます。と言ってもポスター等では地のテキストが既に太めでフォントもやや大きい筈ですから、強調の手法としては枠や下線や網かけに頼らざるを得ません。網掛けや下線は実際にはあまり見かけませんので、ここでは、枠で囲う方法を説明します。

❖ 発表会で見せる OHP の図の文字は、本論文の同じ図の文字より大きくしないと、発表の際、後ろの参加者に見えません。本論文では図中の文字の大きさは地の文の文字の大きさ程度でいいのですが、OHP では上記の理由で旨くありません。また本論文の図を単に拡大しても、文字の大きさの図面に対する比率が上がらない限り有効ではありません。すなわち **OHP 用**には、本論文に載せた図の文字の部分は替えたものを別途用意の方がよいでしょう。

12.1 箱 (ボックス) – box –

L^AT_EX では文字列を箱と考えて横に並べます、そしてそれらをまとめて改めて 1 つの箱として扱い、今度は縦に並べます。それをまた改めて 1 つの箱と考えて... という訳で、この箱はとても基本的な要素なのです。そこで箱を扱うコマンドのうち枠を描くものを使ってみましょう。

12.1.1 単純な箱 : fbox

まずは `\fbox{テキスト}` です。文字列を丁度いい大きさの枠で囲んで表示します。

例題 22

枠付きのボックス命令を使ってみましょう。

```
\fbox{これは fbox です}
{\fboxsep 5mm \fbox{これは fbox です}}
{\fboxrule 1.5mm \fbox{これは fbox です}}\par
\fbox{\fbox{\bf fbox の中に fbox を入れると二重の枠になります}}~
\fbox{$数式も入ります : \int_0^{\infty} \frac{x^2}{1+x^2} dx$}
\begin{center}
\fboxsep6pt\fboxrule0.5mm\fbox{\LARGE\bf タイトルは大きく太字で!}
\end{center}
```

❖ 枠と文字列の空白部分は `\fboxsep` で設定できます。また、枠線の太さは `\fboxrule` で設定できます。`\fbox` と機能も書式も同じですが、枠を表示しないボックス命令として `\mbox{テキスト}` があります。

枠付きの箱 : fbox

これは fbox です

これは fbox です

これは fbox です

fbox の中に fbox を入れると二重の枠になります

数式も入ります : $\int_0^{\infty} \frac{x^2}{1+x^2} dx$

タイトルは大きく太字で!

練習 20

次のような出力を得るソースを考えなさい。(枠線の太さは 0.3mm)

試料の超音波洗浄は以下の行程で行った。

マジックリン: 10分 ⇒ 水道水: 5分 ⇒ アルコール: 15分 ⇒ 純水: 10分

純水洗浄の後は、試料を垂直に立てて自然乾燥を行い、速やかに実験装置の試料ホルダーへと取り付けた。


12.1.2 少し自由な箱 : framebox

文字列の枠に対する位置 (左右、中央揃え) や、枠の横幅を自由に変えることのできるコマンドがあります。`\framebox[幅][位置]{テキスト}` です。

例題 23

`fbox` よりも自由なボックス命令 `framebox` を使ってみましょう。

```
\framebox[10cm][l]{10cmの幅で左揃え}\par
\framebox[10cm][r]{10cmの幅で右揃え}\par
\framebox[10cm][c]{10cmの幅で中央揃え}
```

 `\framebox` と機能は同じですが、枠を表示しないボックス命令として `\makebox[幅][位置]{テキスト}` があります。

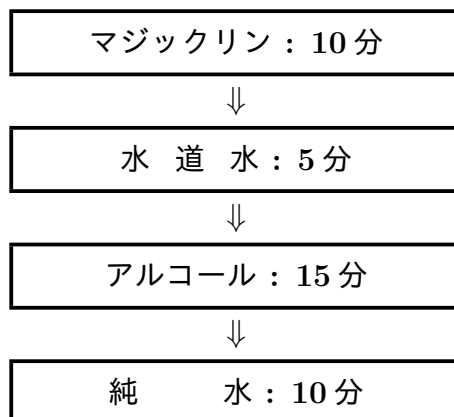
少し自由な枠付きの箱 : framebox

10cm の幅で左揃え
10cm の幅で右揃え
10cm の幅で中央揃え

練習 21

次のような出力を得るソースを考えなさい。(fboxrule=0.5mm, fboxsep=2mm)

試料の超音波洗浄は以下の行程で行った。



純水洗浄の後は、試料を垂直に立てて自然乾燥を行い、速やかに実験装置の試料ホルダーへと取り付けた。


12.1.3 段落が入る箱 : parbox

もう気付いているかもしれませんが、fbox や framebox に入るテキストは改行できません。このような並び方を **LR** モードと呼びます。これでは不便ですから、テキストが改行可能な箱も用意されています。それが `\parbox{幅}{テキスト}` です。ただ枠は付きませんから、さらに fbox で囲んで枠をつけます。

例題 24

改行可能なテキストを入れられるボックス命令 parbox を使ってみましょう。

```
\begin{center}
\large\bf\fboxsep 2mm
\fbox{\parbox{0.8\textwidth}{\いろは\par\いろは}}
\end{center}
いろは
\begin{center}
\fboxsep6pt\fbox{\parbox{8cm}{
\begin{equation}
\cal{A}=\left(\begin{array}{cc}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{array}\right)
\end{equation}}}
\end{center}
```

 幅を指定する括弧が { } となっています。[] ではありませんから間違えないようにしましょう。

段落が入る箱 : parbox

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。
いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

いろはにほへとちりぬるをわかよたれそつねならむういのおくやまきょうこえてあさきゆめみしいねもせず。

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad (6)$$

練習 22

これくらいなら tabular 環境でもできます。確かめてください。

13 罫線

罫線を単に強調や囲い(表としてではなく)のための線分として使ってみましょう。枠囲いがどうもうるさくて好きになれないという人はこちらで工夫しましょう。

13.1 黒塗の箱 : rule

線分という細長いイメージですが、 \LaTeX の罫線 (rule) コマンドはボックス命令と考えられます。つまり細長い箱を線分として使えばいいのです。書式は

```
\rule[リフト]{幅}{高さ}
```

となっています。リフトは箱全体を置く位置を上下するオプションです。

例題 25

黒塗の箱を使って罫線を引いてみましょう。

```
\baselineskip 1.2em
基準線... \rule[0mm]{4cm}{0.3mm}
\rule[5mm]{2cm}{1mm}
\rule[-4mm]{2cm}{4mm}
\rule[-4mm]{3mm}{10mm}
\rule[0mm]{4cm}{0.6mm} ... 基準線

\rule[0mm]{0mm}{4mm} $\Leftarrow$ 幅が0なので見えないが、箱がある。
\begin{center}
テキストの幅一杯の罫線\[-\baselineskip]
\noindent\rule{\textwidth}{0.4pt}
\end{center}
```

黒塗の箱 : rule

基準線...  ... 基準線

← 幅が0なので見えないが、箱がある。

テキストの幅一杯の罫線

13.1.1 自動的に長さが調節される罫線 : hrulefill, dotfill

rule を使った罫線の長さの指定は仕上りをみて自分で調節しなければなりません。自動的に長さを調節してくれる罫線があると便利です。そこで `\hrulefill` の登場です。使い方は簡単なので例題で感じを掴んでください。

例題 26

長さが自動的に調節される罫線を使いましょう。

```
\noindent$\hrulefill{中央}\hrulefill$|\par
\noindent$\hrulefill{中央を長くしました}\hrulefill$|\par
\noindent$\hrulefill{\large\bf 中央をもっと長くしました}\hrulefill$|\par
\noindent$a+b$\dotfill(ix)
```

長さが自動調節される罫線 : `hrulefill`, `dotfill`

中央

中央を長くしました

中央をもっと長くしました

$a + b$(ix)

練習 23

次のようなタイトルを出力するソースを考えなさい。

L^AT_EX 勉強会のお知らせ

練習 24

次のようなタイトルを出力するソースを考えなさい。

■■■■■■ L^AT_EX 勉強会のお知らせ ■■■■■■

14 長い文章の編集

数ページの報告書等は、全文を1つのファイルにするのが当然です。しかし、50ページを越すような論文(まさに修士論文がそうですが)を作成するに当たっては、全文を1つのファイルにするのは不都合が目立ちます。まず、いかに高速なエディタを使っても手直しをしたい場所を見付けるのが大変です。まさか、先頭からスクロールする人はいないでしょうが(セクションのタイトルを検索にかけて飛び、そこからスクロールするのが常套手段でしょう)、とにかく大変です。さらに、これが大きな理由ですが、部分的な手直しを行って、そのレイアウトの確認のために全体を \LaTeX 処理するのは時間の無駄です。そこで、全文を幾つかのファイルに分けて管理することを薦めます。もちろん、 \LaTeX にはそのようなファイル管理を活かす命令が用意されています。

14.1 他のファイルの読み込み: `input`, `include`, `includeonly`

任意の場所で他のファイル”読み込みファイル”を読み込むには単に、

```
\input{読み込みファイル.tex}(.texを省略可能) あるいは
\include{読み込みファイル}(.texを書かない)
```

とします。拡張子”.tex”の扱いより重要な、両者の本質的な違いは、`input`では本当に挿入されるだけですが、`include`では読み込むと同時に改頁されてしまうという点にあります。従って、`include`を使う場合は、ある程度まとまった(章や節)テキスト毎に分ける必要があります。

例題 27

今まで例題や練習で作成した `sample*.tex` や `ex*.tex` を `include`(あるいは `input`) してみましょう。

```
\documentclass[a4j]{jarticle}
\usepackage[graphicx]
\begin{document}
\include{ex1}
\include{ex2}
\include{ex3}
\end{document}
```

このようなファイルを用意して、仮に”`main.tex`”としましょう。この”`main.tex`”をコンパイルします。時間の節約のために不必要なファイルの先頭に%を付ければコメントアウトされ、その部分はコンパイルに掛かりません。逆に、関心のある部分を残して%により他の全てをコメントアウトすればいいのです。

さて、これだけだと `input`の方が自由で、`include`の出番がないようですが、実は `include`には大きな特長があります。それが `includeonly` 命令です。名前から想像がつくように、

```
\includeonly{コンパイルしたいファイル}
```

を、`\begin{document}`より前において、”コンパイルしたいファイル”のみを指定できます。即ち、コンパイルしたいファイルが少ない場合(多くの場合は1つだけ)%で他の部分をコメントアウトするより簡単になります。それよりも重要なことは、コメントアウトするとファイル全体の構造が変化してしまうのですが(章・節・ページ番号 etc.)、この `includeonly`ではその前に \LaTeX が得ていた構造が残ります(具体的には、`*.aux`が上書きされない)。従って、大きく構造が変化しない部分的な手直しにはぴったりの便利な命令なのです。

練習 25

例題 27 において、コンパイルしたいファイルを指定するのに、%で他のファイルを指定から外した場合と、includeonly を使ってそのファイルのみ指定した場合について、結果を比較してみなさい。

14.2 専用マクロの読み込み

同じスタイルの文章を幾つも書く必要がある場合 (学会の投稿論文等はその例) には、マクロを組んで専用スタイルを定義しておき、input すると便利です。そのためには、マクロ定義の知識が必要ですが、それは中級コースでということ、ここでは全体に影響を及ぼすが簡単に設定できる項目を説明します。baselineskip と baselinestretch は文章途中で変えられますが、その他の値はプリアンブルでしか設定できません (即ち、文章途中では変えられません)。

項目	書式例	概要
textwidth	<code>\textwidth=150mm</code>	テキストの幅を設定する。
textheight	<code>\textheight=23.0cm</code>	テキストの高さを設定する。
baselineskip	<code>\baselineskip=18pt</code>	行間の設定。思うようにいかない場合があるので注意。
baselinestretch	<code>\renewcommand{\baselinestretch}{1.5}</code>	baselineskip を標準値の定数倍にします (デフォルトは 1)。こちらの方が旨くゆく場合が多いようです。
topmargin	<code>\topmargin=-10mm</code>	各ページの上端に入れる空白。マイナスなら上にずれることになる。
oddsidemargin	<code>\oddsidemargin=-15mm</code>	ページの左端に入れる空白。twoside では奇数ページにのみ作用 (偶数ページは evensidemargin で設定)、oneside ではすべてのページに作用する。

練習 26

学術雑誌等では、朱筆を入れるためにダブルスペース (1 行置き) を投稿規定に指示している場合があります。そのような場合を想定し、A4 用紙にテキスト幅 13.5cm、行数 25 行でレイアウトする設定を考えなさい。

参考文献

- [1] Donald E. Knuth, *The TeXbook*, (Addison-Wesley Publishing Co., 1984). 邦訳:『TeX ブック』 鷺谷好輝 訳 斎藤信男 監修, (アスキー出版局, 1989)
- [2] 磯崎秀樹, 『L^AT_EX 自由自在』 Computer Today ライブラリ-5, (サイエンス社, 1992)
- [3] J. Kenneth Shultis, *L^AT_EX Notes: Practical Tips for Preparing Technical Documents*, (Prentice Hall, New York, 1994). 邦訳:『L^AT_EX 実用ハンドブック』 河原林美子, 内山厚子 訳 鷺谷好輝, 阿瀬はる美 監修, (プレティンスホール出版, 1995)
- [4] 藤田眞作, 『L^AT_EX マクロの八衢』 Higher Education Computer Series 15, (アジソン・ウェスレイ・パブリッシャーズ・ジャパン, 1995)
- [5] 生田誠三, 『L^AT_EX 文典』 (朝倉書店, 1996)
- [6] 奥村晴彦, 『[改定版]L^AT_EX 2_ε美文書作成入門』 (技術評論社, 2000), <http://www.matsusaka-u.ac.jp/~okumura/texfaq/>
- [7] 中野賢, 『日本語 L^AT_EX 2_εブック』 (アスキー, 1998)
- [8] 乙部巖己, 『pLaTeX2e for Windows Another Manual Vol.1 Basic Kit 1999』 (ソフトバンク, 1999), <http://ms326.ms.u-tokyo.ac.jp/otobe/>
- [9] 内山 孝憲, MacpTeX とその周辺, <http://macptex.appi.keio.ac.jp/~uchiyama/macptex.html>
- [10] 岩熊哲夫, 古川徹生, 『L^AT_EX のマクロやスタイルファイルの利用』 LaTeX のスタイルファイル等に関連して <http://www.civil.tohoku.ac.jp/~bear/node12.html>
- [11] 広瀬雄二, Yet Another L^AT_EX mode for Emacs, <http://www.yatex.org/>
- [12] WinShell, <http://www.winshell.de/>
- [13] 小磯憲史, JTeX Edit, <http://www.math.sci.osaka-u.ac.jp/~koiso/>
- [14] 渡辺純一, Macintosh 版 TeX 統合環境 TeXIE, <http://www.st.rim.or.jp/~jwatanab/texiepage.html>