# 第２回　LaTeXゼミ

森 俊明　　窪田 耕明　　小川 泰正　　吉田 純一

1999年 4月 19日

# 目 次

# 第1章　LaTeXの基礎

前回のゼミでは、LaTeX 環境をインストールする手順を説明しました。今回は実際に LaTeX の文章を書くために必要となる知識を説明します。本章では，LaTeX による文書作成の基本的な手順について説明します．前回の課題で使用されていた命令については本章で解説しています。

## 1.1　LaTeXでの約束ごと

LaTeX で文書を作成するときに注意しなければならない約束ごとについて説明します.

### 1.1.1　ファイル名

LaTeX に限らず，TeX では，原稿ファイルの拡張子を ”.tex” にするという約束があります．本当は ”.tex” 以外でも別にかまわないのですが，あえて約束違反をする必要もないでしょうし，ファイルの種類がわかりやすいこともありますので，必ず ”.tex” にして下さい.

### 1.1.2　最低限のルール

1. 「TeX や LaTeX の組版命令は，原則として"\" で始まり，原則として半角空白でおわる」
   TeX や LaTeX の組版命令は，そのほとんどが"\" で始まります．組版命令に続く文字が，全角の空白や句読点，全角および半角の括弧や記号（@など），半角の数字，TeX や LaTeX の命令であれば，半角の空白は必要ありません.

2. 「半角の空白はいくつ続いても 1 つとみなされる」
   "\TeX "と書いた場合も"\TeX　"と書いた場合も出力上は同じ結果となります．なお改行直後（行頭）の半角空白は無視されます.

3. 「全角文字の直後で改行すると改行は無視され，半角文字の直後で改行すると改行は空白とみなされる．また，2 つ以上続いた改行（改行コードのみの行）は，段落の切れ目として扱われる.」
   「空白」のところで詳しく説明します.

4. 「半角の記号文字の中には，そのまま入力しても出力できない特殊な文字がある」
   次の半角記号は，TeX や LaTeX ではそのまま入力しても出力できません.
   `#`, `$`, `%`, `&`, `_`, `{`, `}`, `<`, `>`, `\`, `|`, `^`, `~`
   これらの記号を出力する一番手軽な方法は，全角文字を使用することです．全角文字は TeX の内部で特殊な意味を持っていませんので，自由に使用することができます.

5. 「半角のカタカナは使用できない」
   TeX や LaTeX では，半角のカタカナを入力してもエラーにはなりませんが，それを出力できる

DVI ドライバがありません．

6. 「”%”以降は，改行コードも含めてコメントと見なされる」
   行中の”%”記号以下は改行コードも含めてコメントとして扱われます．このため，実際の原稿ファイルでは改行しているのに，改行していないのと同様に処理したい場合などには”%”記号を利用して改行します．

7. 「LaTeX の原稿には，必ず記述しなければいけない命令がある」
   LaTeX の原稿には，`\documentclass`，`\begin{document}\end{document}` という命令が記述されていなければなりません．

### 1.1.3 LaTeX 原稿の中身

簡単な原稿を例にとって、LaTeX　原稿の中身について説明します。次のような原稿ファイルがあるとします．これは、Hello, TeX と表示する簡単な原稿です。

```
1:  \documentclass[a4paper]{jarticle} [1]
2:  \begin{document}
3:  Hello, \TeX !
4:  \end{document}
```

`1:  \documentclass[a4paper]{jarticle}`

\記号で始まっているので，TeX や LaTeX の命令だとわかります．今のところ，これは LaTeX で文書を作成する場合の「おまじない」だと思って差し支えありません．この行の意味を簡単にいうと，これから記述する文章の種類と文書の書式を LaTeX に指定しています．”[a4paper]”は「A4 判の用紙に組み版せよ」ということです．`"{jarticle}"`は「日本語が使用されるかもしれない比較的短い文章である」ということを意味しています．これらの文字列は，LaTeX に対する組み版命令ですから，当然できあがった文書には出力されません．

`2:  \begin{document}`
`4:  \end{document}`

2行目の`"\begin{document}"`は「ここから本文が始まる」と，4行目の`"\end{document}"`は「ここで本文はおしまい」と LaTeX に教えています．LaTeX の原稿には`"\begin{document}"`と`"\end{document}"`という 1 組が必要で，このあいだに本文を書くという決まりがあります．

`3:  Hello, \TeX !`

実際に，プレビューしたときには，この行に似た一文が表示されます．`"\TeX"`の部分に注目してください．これは，TeX というロゴを出力するための命令です．

## 1.2 空白・改行・段落

TeX では，たいていの場合において，単語間の空白や段落の最初の空白を自動的に空けてくれます．ここでは，TeX が自動的に空けてくれる空白の規則について説明します．

[1] バックスラッシュは実際はすべて半角の”¥”です．これは標準で TeX が利用できるフォントのなかに半角の”¥”記号が含まれていないためです．

### 1.2.1 自然な空白

#### 英文の場合

TEX では英文の単語や文の区切りで自動的に適量の空白を空けるようになっています．このとき，単語の区切りとしての空白より，文の区切りとしての空白の方が幅広く確保されます．文の区切りでは，ピリオドを検知し，ある処理（詳しくは参考文献を参照してください）を行って余分な空白が空けられます．しかし，この規則では，”Dr. Knuth”（本当は”Dr. Knuth” となって欲しいのですが）は，間違って文の区切りとして扱われてしまいます．このような場合の対処については，参考文献に詳しく説明してありますので参照してください．

また，先に述べましたように，TEX や LATEX の命令のあとには，原則的に命令の終わりを示すために半角の空白を挿入しておかなければなりません．したがって，TEX や LATEX の命令の直後に半角空白を空けても，単語間の自然な空白は空きません．たとえば，`"\TeX for Windows"`と記述した場合の出力は，”TEXfor Windows” となってしまいます．このような場合，TEX に半角空白以外の方法で命令の終端を認識させるか，既に述べた単語間の空白を空ける命令を使用して対処します．具体的には，次の 5 つの方法で対処することができます．

1. `\TeX\ for Windows`
2. `\TeX~for Windows`
3. `\TeX\@for Windows`
4. `\TeX{} for Windows`
5. `{\TeX} for Windows`

それぞれの詳しい意味は参考文献を参照してください．

#### 和文の場合

和文のみの文書，あるいは英文と和文が混在している場合には，空間規則が若干変化します．

```
1:  \documentclass[a4paper]{jarticle}
2:  \begin{document}
3:  コンピュータ社会で技術的な貢献をした人には ACM
4:  Turing 賞が贈られます．特に，コンピュータの中心
5:  分野に，著しい影響を及ぼすような貢献が対象とされています．
6:  \end{document}
```

この文章を TEX で処理すると，次のように出力されます．

コンピュータ社会で技術的な貢献をした人には ACM Turing 賞が贈られます．特に，コンピュータの中心分野に，著しい影響を及ぼすような貢献が対象とされています．

まず，3 行目末〜4 行目の「ACM Turing 賞」の部分と，4 行目末〜5 行目の「著しい影響」に相当する部分に注目してください．行末が半角文字の場合には，改行が単語間の空白として処理されます．これに対して，行末が全角文字の場合には，特に何の処理もされません．

### 1.2.2 段落の区切り

TEX は，連続する改行記号を段落の区切りとします．簡単にいえば，何も記述されていない改行記号だけの行があれば，それを段落の区切りとみなします．段落の始まりでは，TEX が自動的に，英文の場合には適当な分量だけ，和文の場合には全角 1 文字分の字下げをしてくれます．

また，ワープロの場合と異なり，改行コードだけの行を続けても，縦方向の空白を空けることはできません．縦方向の空白を空ける場合には，そのための命令を使用します．

### 1.2.3 書いたまま出力する方法

先に述べました通り，次に示す文字は，TEX や LATEX で特別な役割を果たしますので，原稿中にそのまま記述しても出力することができません．

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

したがって，"(^_^)"と書くとエラーになります．これらの特殊文字を出力する最も簡単な方法は，"\verb"という命令を使用することです．

```
\verb|(^_^)|
```

この命令は，引数の始まりと終わりに同じ区切り記号を（上の場合は|）を指定することで，区切り記号のあいだに記述された原稿をタイプライタ文書でそのまま出力する働きを持っています．区切り記号のあいだに記述する文字列は基本的にはどのようなものでもかまいませんが，両端の区切り記号と同じ文字だけは使用できません．また，*記号を両端の区切り文字として使用することもできません．これは，\verb*という別の命令があるからです．

# 第２章　基本的な文書の編集

## 2.1　文字の修飾

文章を書くときに必要になってくる文字の修飾方法は LaTeX にも当然あります。ここでは、書体や文字サイズの変更方法について簡単にお話します。

### 2.1.1　文字サイズの変更

文字の大きさを変えるには、通常は次の 10 種類の命令を使用します。出力した文字の右側に、その出力方法を記述しておきます

sample {\tiny sample}

sample {\scriptsize sample}

sample {\footnotesize sample}

sample {\small sample}

sample {\normalsize sample}

sample {\large sample}

sample {\Large sample}

sample {\LARGE sample}

sample {\huge sample}

sample {\Huge sample}

### 2.1.2　書体の変更

書体の変更方法は以下の通りです。

**sample**\textbf{sample}

sample{\rmfamily sample}

sample{\sffamily sample}

sample{\ttfamily sample}

sample{\mdseries sample}

**sample**{\bfseries sample}

sample{\upshape sample}

*sample*{\itshape sample}

*sample*{\slshape sample}

SAMPLE{\scshape sample}

## 2.2 見出し

見出しは、文章のまとまりを階層的に表現するという点で、LaTeX においては非常に重要な役割を果たします。LaTeX では、見出しの始まりを次の命令で認識します。上にある命令ほど大きな文章単位です。

| | |
|---|---|
| \section{title} | 節 |
| \subsection{title} | 項 (小節) |
| \subsubsection{title} | 目 (少々節) |
| \paragraph{title} | 段落 |
| \subparagraph{title} | 小段落 |

これらの命令には、引数としてそれぞれの見出しのタイトルを指定します。これらの命令を使用すると、LaTeX はクラスファイルに定義されている通り、番号付けや見出しの大きさなどを自動的に決定します。

この他にも\part{title}(部) や\chapter{title}(章) といった見出しもありますが、これは jarticle クラスでは使用できません。

## 2.3 改ページと空白の開け方

### 2.3.1 改ページ

改ページは、通常 LaTeX が自動的に位置を決めて行います。しかし、ユーザが自ら指定することも可能です。次に改ページに関するコマンドをいくつか記します。

| | |
|---|---|
| \newpage | 強制的に改ページしたい場合には、その場所で\newpage 命令を使用します。二段組みをしている場合で、現在左の段であれば、\newpage 命令によって右側の段に移動します。 |
| \clearpage | 強制的に改ページしたい場合には、その場所で\clearpage 命令を使用します。\newpage 命令との違いは、\clearpage 命令の使用時に配置が決定されていない図表があれば、それらを全て出力してから改ページされることです。また、2 段組であるか無いかに関わらず、常に新しいページを起こします。 |
| \cleardoublepage | 次のページが右ページから始まるように、必要に応じて白紙のページを挿入して改ページしたい場合には、その場所で\cleardoublepage を使用します。 |

### 2.3.2 空白の開け方

先ほど説明しましたように，TEX では空白がいくつ続いても 1 つの空白とみなされます．1 つ以上の空白を空けるには，次の命令を使用します．

```
\vspace{height}
\hspace{weight}
```

\vspace{height}命令は”height”に記述された値だけ行間を空け，\hspace{weight}命令は”weight”に記述された値だけ文字と文字との間隔をあけます．

\vspace{10mm}
この命令によって，

というように，行間を設定することが可能です．

\hspace{30mm}という命令では，　　というように文字と文字の間隔を設定することができます．

また，段落間に適当な大きさの空白をあけたい場合には，次のような命令を使用します．

\smallskip は小さな空白を作ります．

\medskip は中くらいの空白を作ります．

\bigskip は大きな空白を作ります．

このように，段落間の改行を操作することができます．

## 2.4 注釈

脚注を使用したい場合には、\footnote 命令を使用します。\footnote 命令を使用すると LATEX は組版時に自動的に番号を付け、引数に指定された文字列をページの下部に脚注として出力します。また、脚注でなく、本文の隣に傍注を出力することもできます。傍注を出力したい場合には、\footnote 命令の変わりに、\marginpar 命令を使用します。この時、傍注欄の横幅を指定することもできます。

注釈はこのように、横に表示することも可能です。

## 2.5 文字の位置

LATEX でもワープロソフトと同じように文字の中央寄せ、左寄せ、右寄せを指定することが可能です。それぞれ center 環境、flushleft 環境、flushright 環境となります。

center 環境

flushleft 環境

flushright 環境

## 2.6　環境と命令

LaTeX の最大の特徴は、論理デザインが可能であるということです。論理デザインでは、中央揃えであるとか、段落であるとか、箇条書きであるとか、表であるといった、文章の論理的な構成を、本文とは別にソースの中に書き加えなくてはなりませんが、それによって、視覚的なデザインを意識せずに文書を作成することができるのです。

LaTeX の場合、文章の論理的な構成は「環境」と呼ばれるマクロ命令を通して TeX に伝えられます。もっとも代表的なものは、本文の始まりと終わりを宣言するための document 環境があります。この環境は、この２つで囲まれた範囲が、本文であるということを示しています。

環境は、

¥begin（環境名）
¥end（環境名）

という書式で記述し、begin と end で挟まれた部分がその環境になります。このほかにも、中央揃えにする center 環境や、図版を張り込むための figure 環境などが用意されています。

LaTeX では環境の他にも命令と呼ばれるものが用意されています。節を定義する¥section 命令などがそれにあたります。命令は、

¥命令 {(引数)}

という書式で記述します。¥section はじめにという命令の場合、section という命令に「はじめに」が引数として渡されることで、「はじめに」という節があるということを宣言します。

# 第3章　数式モード

LaTeX は通常,「段落モード[1]」と呼ばれるモードで作動しています. 他に表などを作成するときのモードとして「LR モード[2]」と呼ばれるモードがあります.LaTeX で数式を入力する場合には，もう１つの数式モードと呼ばれるモードを使用します.

## 3.1　数式のアレンジ

ここでは、数式モードでの数自体をアレンジする主な方法を述べていきます。

### 3.1.1　空白

数式モードでは，TeX や LaTeX の命令の区切りを表す以外の半角空白は無視されます. それでは，数式中で空白を制御する場合にはどうすればいいかというと，次の表 3.1 ような命令を使用します.

表 3.1: 数式モード内での空白制御命令

| 命令 | 空白の大きさ | 数式モード以外での使用 |
|---|---|---|
| \ | 半角の空白 | 可 |
| \quad | 全角の空白 (クワタ) | 可 |
| \qquad | ２個のクワタ | 可 |
| \, | 細スペース (クワタの 1/6) | 可 |
| \> | 中スペース (クワタの 2/9) | 不可 |
| \; | 大スペース (クワタの 5/18) | 不可 |
| \! | 負の細スペース (クワタの-1/6) | 不可 |

### 3.1.2　文字列の扱い

数式モードに記述された文字列は，原則として全て数式 (変数) とみなされ，数式用に用意された特殊なイタリック体 (数式イタリック体) で出力されます. たとえば”$diff\ (a)$”という出力を得たい場合に，”$diff(a)$”と入力すると，”d”,”i”,”f”,”f”はそれぞれ個別の変数として扱われ，”$diff(a)$”と出力されてしまいます. このようなときに，本文中のイタリック体と同様の出力を得るためには，”`$¥mathit{diff}(a)$`"と書きます.3.2

[1]段落モードとは，単語や文など，原稿を文章の固まりとして取り扱い，行や段落，ページに分割していくモードです.
[2]LR モード中では，改行が起こりません.

表 3.2: $diff(a)$ の出力

| 入力 | 出力 | 数式モード |
|---|---|---|
| diff(a) | diff(a) | × |
| `$diff(a)$` | $diff(a)$ | × |
| `$\mathit{diff}(a)$` | $\mathit{diff}(a)$ | ○ |

### 3.1.3 添え字

ここでは，数式の添え字について説明を行います．下添え字は"_"上添え字は"^"記号に続けて書きます．これらは同時に指定することが可能であるため，数式中で `a^i_j` と書けば $a^i_j$ となります．また，添え字を利用することにより，積分の範囲指定なども書く事ができ，`\int_a^b` と書けば $\int_a^b$ となりますし，`\sum_{n=1}^{100}`と書けば $\sum_{n=1}^{100}$ と表示されます． また，添え字は，入れ子にすることも可能です．$x^{y^z}$ と出力したければ，`$x^{y^z}$`と書きます．ここで，`$x^y^z$`と書いてしまうと，TEX が添え字の順番が分かずにエラーを出すので，添え字の順番ははっきりとわかるように記述しておかねばなりません．

## 3.2 基本的な数式環境

LATEX には，さまざまなバリエーションの数式を出力することができるように，いくつかの数式環境が用意されています．ここでは比較的使用度の高い数式環境について説明します．

### 3.2.1 math 環境

math 環境は，文中に数式を出力するための環境です．たとえば，次のようになります．

**例** 二次関数 $y = x^2$ のグラフは放物線である．

しかし，文中でいちいち math 環境を使用すると，文章のつながりがよく分からなくなるので，通常は，$と$で数式を囲んで使用します．

### 3.2.2 displaymath 環境

displaymath 環境は，数式を別の行に出力したい場合に使用します．同様の働きをする命令として，¥[ と ¥] があります．

**例** 以下に示すのが，displaymath 環境の出力例である．

$$y = x^2$$

### 3.2.3 equation 環境

数式に参照用の番号をつけたい場合は equation 環境を使用します. 基本的には，displaymath 環境と同じです.

**例** 二次関数

$$y = ax^2 + b^2 + c \tag{3.1}$$

を微分すると，

$$y = 2ax\,dx + b\,dx \tag{3.2}$$

となる.

### 3.2.4 eqnarray 環境

複数の数式を連続して出力したい場合には，eqnarray 環境を使用します. 環境内の各行には自動的に番号が付けられます．もし，どの行にも番号を付けたくないなら，環境名を eqnarray*としてください.

**例**

$$y = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \tag{3.3}$$

$$y = \sin x \tag{3.4}$$

$$y = \cos x \tag{3.5}$$

$$y = \tan x \tag{3.6}$$

$$y = e^x \tag{3.7}$$

$$y = \log x \tag{3.8}$$

このソースを見てもらうと，各行の最後に改行の命令 ~~¥¥~~がついているのがわかります.eqnarray 環境では，array 環境[3]と同じく自動的に改行を行わないため，ユーザが改行命令を用いて改行を行う必要があります.

また，最後の行に改行命令をつけてはいけません. もしここに改行命令を入れると，一番下の行のもう１段下に空白の行が現れ，その行にも参照用の番号が付いてしまうからです.

eqnarray 環境は，array 環境に似ていることから，上の文に少し変形を加えるだけで，次のような出力を得ることができます.

**例**

$$\begin{aligned} y &= a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 && (3.9)\\ y &= \sin x && (3.10)\\ y &= \cos x && (3.11)\\ y &= \tan x && (3.12) \end{aligned}$$

[3]表や行列を作成するための環境で，これを使用すると LR モードになります.

$$y = e^x \tag{3.13}$$
$$y = \log x \tag{3.14}$$

ソース中の"&"記号は列の要素をそろえるための命令です．

また，1 行におさまらない式を 2 行に分割して出力するには，次のようにします．

**例**

$$z = xy \tag{3.15}$$
$$y = x^2 + xy + -y^2 + x - y - 1 \tag{3.16}$$

"+ x - y - 1"の前に {} が記述されているのに注意してください．TEX は数式中の空白をすべて自動的に調整するので，たとえば"+1"と"a+1"’では"+"の後の空白が微妙に違ってきます．上の例の場合，"+ x - y - 1"の"x"の前の"+"は符号ではなく二項演算子なので，TEX にその事を伝えるために，空白を入力しているのです．

ここで，もう 1 つ重要なことがあります eqnarray 環境は環境内の各行に番号をつけてしまうので，多行にわたる数式を記述するためには，数式番号を制御するための命令が必要になってきます．それを行うのが ¥nonumber 命令で，これをその行の改行の前に記述しておけば，その行に番号が振られることはありません．

## 3.3 相互参照

図表や式などに番号をつける理由の 1 つとして，"詳細は図 3 参照"などのように参照箇所を示すケースが挙げられます．この場合，"3"などの直接的な番号指定は避けるべきです．文章の変更によって"4"になる事もあるからです．LATEX ではこのような方法の代わりに，任意のラベルを図に割り当てておき，そのラベルを参照して図の番号を生成する事ができます．ラベルの割り当てには ¥label コマンドを，参照には ¥ref コマンドを使います．通常の文章中で ¥label を使用すると，そのラベルには現在の見出しの番号が割り当てられます．以下の例では，コマンド ¥label{eq:euler}によって式番号にラベル eq:euler が割り当てられ，¥ref{eq:euler}でその番号を参照しています．

**例** 式 6.1 をオイラーの公式と呼ぶ．

$$e^{ix} = \cos x + i \sin x \tag{3.17}$$

### 3.3.1 数式の例

ここでは，比較的使用頻度が高そうなサンプル数式を示しておきます．ソースと比較して数式を書くときの参考にしてください．

例 1： $y = \dfrac{1}{x+1}$

例 2： $\int_a^b f(x)\,dx$

例 3： $df = \dfrac{\partial f}{\partial x}\,dx + \dfrac{\partial f}{\partial y}\,dy$

例 4： $\displaystyle\int_0^1 f(x)\,dx = \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{k=n-1} f\left(\frac{k}{n}\right)$

例 5： $e^{\sqrt{-1}\,nx} = \cos nx + \sqrt{-1}\,\sin nx$

例 6： $\langle f, g\rangle = \displaystyle\int_x \bar{f} g\,d\mu \quad \text{in } L^2(X, d\mu).$

例 7： $(df)_p\left(\left.\dfrac{dc}{dt}\right|_{t=0}\right) = \left.\dfrac{d(f\circ c)}{dt}\right|_{t=0}$

### 3.3.2 数式を使おう！

数式を扱う機会は多いと思いますので，ぜひマスターしてください．前回と同じように，最後にこのレジュメのソースを添付しておきますので，参考にしてください．

課題として，次のページと同じものを LaTeX で作成してください．提出期限は，第 3 回 TeX ゼミが始まるまでです．

# 第4章　表の作成

## 4.1　table 環境

table 環境とは，表を作成するための領域（表の配置）を確保する命令です．具体的には表 6.8「table 環境の使用例」を見て下さい．　￥caption 命令と￥label 命令については，数式のレジュメを参照して下さい．

## 4.2　tabular 環境

tabular 環境とは表組みを作成するための環境です.tabular 環境を使用すると，かなりこった表を作成する事も可能ですが，今回のゼミでは表中での文字の揃え方や，罫線の引き方など基本的なものだけにとどめておきます．また,table 環境と tabular 環境を併用して利用する事で表の配置と作成を同時に出来る環境が整うのです．

まずは，表 4.2 罫線の無い表の例」にもっとも基本的な表組みの例を示します．

表 6.6 罫線の無い表の例のソース」を見て下さい．最初の１行目は table 環境の宣言です．２行目は，表を中央に揃えるための center 環境の宣言です．そして，３行目が表を作成する tabular 環境の宣言となっています．後ろにあるブレースに囲まれた４つの文字は，表の列の数と，その列の要素の配置位置を指定しています．それぞれ表 4.4「tabular 環境の引数の説明の表」のような意味が有ります．

表 4.2 の場合は，左から３列が左揃えで，一番右の列が中央揃えになっています．この表 6.6 において４行目から下４行が，実際の表を構成している要素です．また表 6.6 中の１行が表の１行に対応しています．１行目の各要素のように，文字を太くする事も可能です．要素同士の境界は,&で指定します．行末に書かれている￥￥は，行と行との間を表わす命令であり，改行命令では有りません．よって表の最後の行にはつけてはいけません．

表 4.5 に「罫線のある表の例を示します．

表 6.6 罫線の無い表の例のソース」を見て下さい．最初の１行目は table 環境の宣言です．２行目は，表を中央に揃えるための center 環境の宣言です．そして，３行目が表を作成する tabular 環境の宣言となっています．後ろにあるブレースに囲まれた４つの文字は，表の列の数と，その列の要素の配

| |
|---|
| ￥begin{table} |
| 実際の表を作成する場合には, ここに tabular 環境を記述する. |
| ￥caption[短い説明]{長い説明} |
| ￥label{ラベル} |
| ￥end{table} |

表 4.1: table 環境の使用例

| 書名 | 著者 | 出版社 | 定価 |
|---|---|---|---|
| TEX ブック | Donard.e.Kunth | アスキー出版局 | 6,000 円 |
| 文書処理システム LATEX | leslei Lamport | アスキー出版局 | 2,800 円 |
| LATEX 美文書作成 | 奥村晴彦 | 技術評論社 | 2,500 円 |

表 4.2: 罫線の無い表の例

```
￥begin{table}
￥begin{center}
￥begin{tabular}{lllc}
￥bf{書名}&￥bf{著者}&￥bf{出版社}&￥bf{定価}￥￥
￥TeX ブック&Donard.e.Kunth&アスキー出版局&6,000 円￥￥
文書処理システム￥LaTeX &leslei Lamport&アスキー出版局&2,800 円￥￥
￥LaTeX 美文書作成&奥村晴彦&技術評論社&2,500 円
￥end{tabular}
￥end{center}
￥caption{罫線の無い表の例}
￥label{hyou1}
￥end{table}
```

表 4.3: 罫線の無い表の例のソース

| 文字 | 要素の配置 |
|---|---|
| l | 左揃え |
| c | 中央揃え |
| r | 右揃え |

表 4.4: tabular 環境の引数の説明

| 書名 | 著者 | 出版社 | 定価 |
|---|---|---|---|
| TEX ブック | Donard.e.Kunth | アスキー出版局 | 6,000 円 |
| 文書処理システム LATEX | leslei Lamport | アスキー出版局 | 2,800 円 |
| LATEX 美文書作成 | 奥村晴彦 | 技術評論社 | 2,500 円 |

表 4.5: 罫線のある表の例

置位置を指定しています. それぞれ表 4.4「tabular 環境の引数の説明の表」のような意味が有ります. 先ほどの表 6.6「罫線の無い表の例」と見比べて下さい.tabular 環境の開始の行に書かれているブレースの中身が少し違う事に気づくでしょう. 先ほど説明した要素の配置を決める引数の間に,| が入っています. これが, 表の縦方向の罫線の挿入位置を決めています. 列と列の間に 2 本の線を引きたければ,|| のように 2 個続けて入力して下さい.

横方向の罫線を引く場合には, 行と行との間 (つまり￥￥命令の直後) で￥hline 命令を挿入して下さい. 縦方向の線と同様に, 横線を 2 本引くには￥hline 命令を 2 回続けて書いて下さい.

# 第5章　行列の作成

LaTeX には行列を作成する方法がいくつか有りまあす. ここでは, もっとも基本的であるといえるに array 環境ついての説明をします.

## 5.1　aray 環境

array 環境とは, 配列を作成するための環境です. 数式モード 内で使用され, 主に行列を作成するのに使用されます. 式（5.1）に, 最も簡単な例を示します.

$$\begin{array}{cc} a & b \\ c & d \end{array} \tag{5.1}$$

式 (5.1) を見て下さい. 先ほど説明した tabular 環境に, 非常に似ています. 実際, 引数の指定方法や￥￥の使用方法も同じで,| や￥hline 命令を使用してもかまいません. 違うのは, 参照のための番号の位置くらいです. しかし ,tabular 環境と同じ使い方をしていては意味が無いので, まずは括弧の付け方を覚えて下さい.

$$\left( \begin{array}{cc} a & b \\ c & d \end{array} \right) \tag{5.2}$$

行列 (5.1) との違いは, 括弧を作る命令￥left(と\verb￥right|が付け加えられています. 例えば,￥left(は ( を作成し,￥right) は ) を作成します. 括弧の大きさは,TeX が自動的に調整してくれます.￥array 命令の外で使用している事からも分かるように, この命令は行列だけでなく, 普通の数列に対しても使用可能です.　￥left 命令と￥right 命令を使用する時には, 必ず￥left 命令と￥right 命令の数がつりあっていなければなりません.

```
￥begin{equation}
￥begin{array}{cc}
a&b￥￥
c&d
￥end{array}
￥label{gyou1}
￥end{equation}
```

表 5.1: aray 環境 の使用例 1 のソース

しかし，数が揃っていればどのような括弧がきてもいいので，次のような出力を得る事も可能です．

$$\left( \begin{array}{cc} a & b \\ c & d \end{array} \right\{ \tag{5.3}$$

式 (5.3) を利用して，場合分けを書く事が可能です．

$$|x| = \left\{ \begin{array}{cc} x & (x > 0) \\ -x & (x < 0) \end{array} \right. \tag{5.4}$$

￥left 命令と￥right 命令の数はつりあっていなければならないので，式の右側に見えない括弧を出力する命令が必要になってきます．それが，￥right. で表わされています．

aray 環境は数式モードの中でなら，何度でも宣言可能なので，次のように書く事も可能です．

$$\left( \begin{array}{cc} a & b \\ c & d \end{array} \right) \left( \begin{array}{cc} p & q \\ r & s \end{array} \right) = \left( \begin{array}{cc} ap + br & aq + bs \\ cp + dr & cq + ds \end{array} \right) \tag{5.5}$$

# 第6章　図版の貼り込み

## 6.1　figure環境

figure 環境は，図版を貼り込む領域を確保する環境です．具体的には次のように指定します．

```
￥beginfigure
  実際の図版
  ￥caption[短い説明] 長い説明
  ￥label ラベル
￥endfigure
```

3行目の￥caption 命令では，図版の説明文（キャプション）を指定します．{長い説明} に指定したことが，実際の図版の下に出力されます．また，目次に図目次を加える場合，長い説明がそのままリストアップされます．この図目次に長い説明の代わりに別の短い説明をリストアップさせたい時は，[短い説明] を指定してください．4行目の￥label 命令では，任意のラベルを図に割り当てておき，そのラベルを参照して図の番号を生成することができます．例えば“ 図2に示すように ”などのように直接的な番号指定をしてしまうと，後から文章の変更によってそれが図3になることも十分ありえるのです．そのような手間を省くために￥label 命令をすると考えていいでしょう．ラベルの割り当てには￥label コマンドを，参照には￥ref コマンドを使います．以下の例では，コマンド￥labeleq:euler によって式番号にラベル eq:euler が割り当てられ，￥refeq:euler でその番号を参照しています．

**例**　式6.1をオイラーの公式と呼ぶ．

$$e^{ix} = \cos x + i \sin x \tag{6.1}$$

## 6.2　graphicパッケージ

ここでは，画像ファイルを文書の中に貼り込む方法を説明します．

### 6.2.1　graphicパッケージの読み込み

graphic パッケージを読み込むには，プリアンプルで￥usepackage 命令を使用して読み込みます．具体的にはプリアンプルで

```
￥usepackagegraphics
```

というように指定します．

### 6.2.2 画像ファイルの貼り付け

画像ファイルを貼り付けるには，¥includegraphics という命令を使用します．¥includegraphics 命令は次のような書式で使用することで，画像ファイルを *file*，幅 *width*，高さ *height* の大きさで貼り付けます．

¥includegraphics[*width*,*height*]*file*

ファイル名の指定

¥includegraphics 命令の引数となる *file* の拡張子は，必ず小文字で記述してください．これは，画像ファイルの形式が何であるのかを LaTeX に伝えるためです．

出力サイズの指定

幅 *width* や高さ *height* は，画像ファイルを張り込んだときの大きさを表します．これらの値を指定すると，DVI ドライバが画像ファイルを *width* や *height* の大きさになるように拡大または縮小して出力します．

## 6.3 仮想プリンタの作成

### 6.3.1 なぜ仮想プリンタが必要か？

私たちが普段パソコンで扱っている絵（画像）は，大きく 2 つに分かれます．一つは Paint 系（ビットマップ，ペイントで作る絵，etc）で，もう一つは EXCEL や WORD などの draw 系です．ps(eps) 形式は draw 系に属します．しかし例えば，Excel でグラフを作成してもそれを eps 形式で保存することはできません．しかし仮想プリンタを使用することにより eps 形式に保存できるのです．（方法は後で詳しく述べます．）また，文字情報を Post Script 形式にするためには，PS プリンタの持っているフォントを使用する必要があることも，仮想プリンタが必要な理由の一つになっています．

### 6.3.2 仮想プリンタの作成手順

まず，ps424jpn を開きます．そしてその中の setup.exe を実行します．PostScript プリンタをシステムに追加しますといわれるので，次へを押します．セットアッププログラムのインストールでは，いいえを選択して次にいきます．プリンタのタイプですが，図 6.1 のようにローカルプリンタを選択して次にいきます．

次の，PPD による PostScript プリンタのインストールでは，図 6.2 のように c:¥の WINDOWS の SYSTEM の中の Acrobat Distiller 3.0J を選択します．

次のローカルポートの選択では，図 6.3 のようにファイルを選択して次にいきます．

プリンタの追加では，図 6.4 プリンタ名を入れ（仮想プリンタと入れると分かりやすい）2 つの選択肢ともいいえを選んで次にいきます．

次に追加するプリンタのプロパティを聞いてくるので，図 6.5 のようにフォントの中の「フォント置き換えテーブルを使用」を選択しておきます．後はそのままでいいです．

次のファイルへ出力というところはそのままでキャンセルをしておきましょう．以上でセットアップが終了です．うまくできていればスタートメニューの設定の中のプリンタを見てみると作成した仮

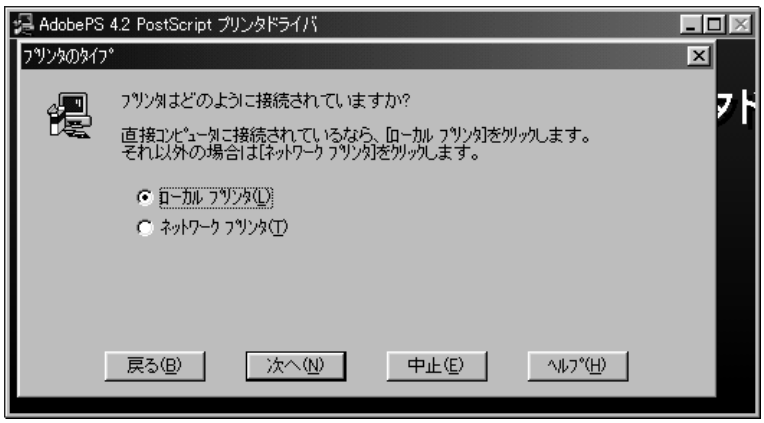


図 6.1: 仮想プリンタの作成

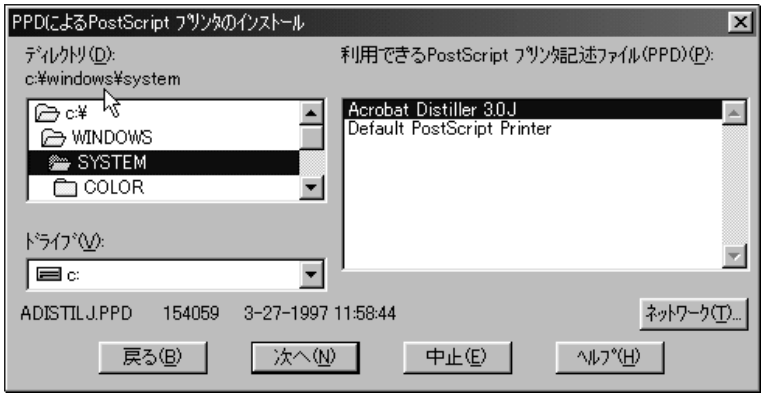


図 6.2: 仮想プリンタの作成


図 6.3: 仮想プリンタの作成

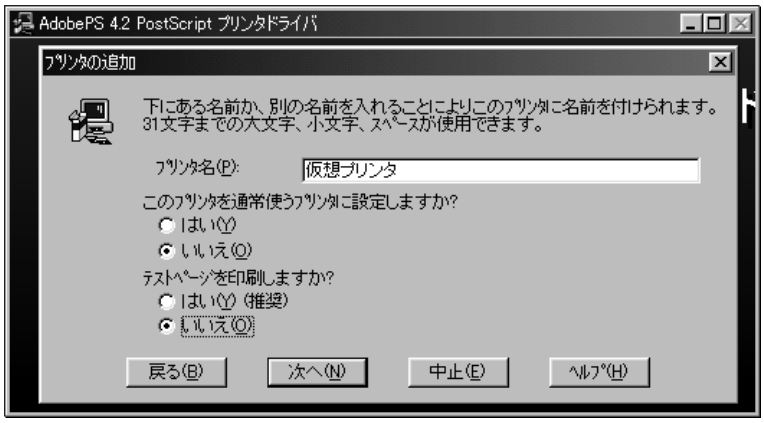


図 6.4: 仮想プリンタの作成

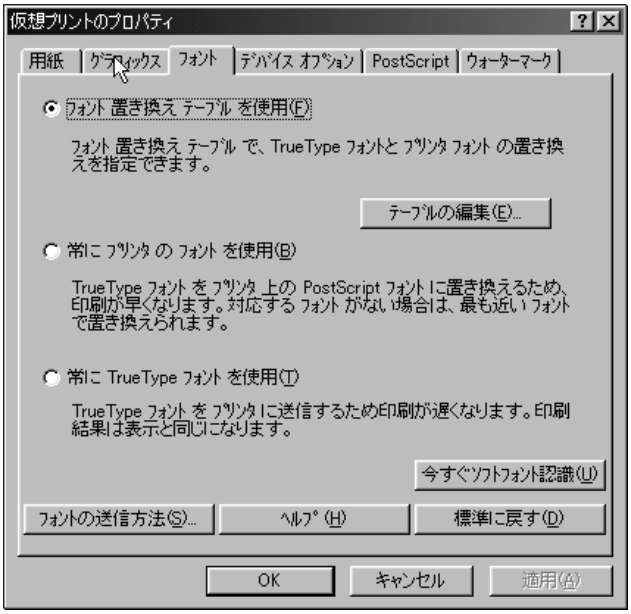


図 6.5: 仮想プリンタの作成

想プリンタが入っているはずです．ではこの仮想プリンタを使って draw 系のファイルを eps ファイルにしてみましょう．

## 6.4 画像ファイルの作成

実際に画像ファイルを作成してみます．

### 6.4.1 EXCEL によるグラフ作成

まず，EXCEL でグラフを作成することをやってみましょう．はじめに，Microsoft Excel を立ち上げます．そこで実際にグラフを作ります．そしてこれを eps 形式で保存します．メニューのファイルから印刷を選んで，プリンタ名に「仮想プリンタ」を選び，プロパティの「PostScript でカプセルかされた PostScript（EPS 形式)」を選び，OK を押します．この作業を行うことにより，使えるフォントの中に PS プリンタのフォントが含まれていることでしょう．図 6.6 のようにグラフエリアの書式設定を選び図 6.7 のようにフォントを見れば，聞いたことのないフォントが追加されているのに気がつくでしょう（普通は，中ゴシック BBB を使うのが好ましい)．そのフォントでグラフを作成したら，メニューの印刷を選んで，今度は実際に印刷します．するとファイル名を聞いてくるので，そこで自分のファイル名に EPS の拡張子をつけましょう．これで，指定した送り先に EPS ファイルができています．

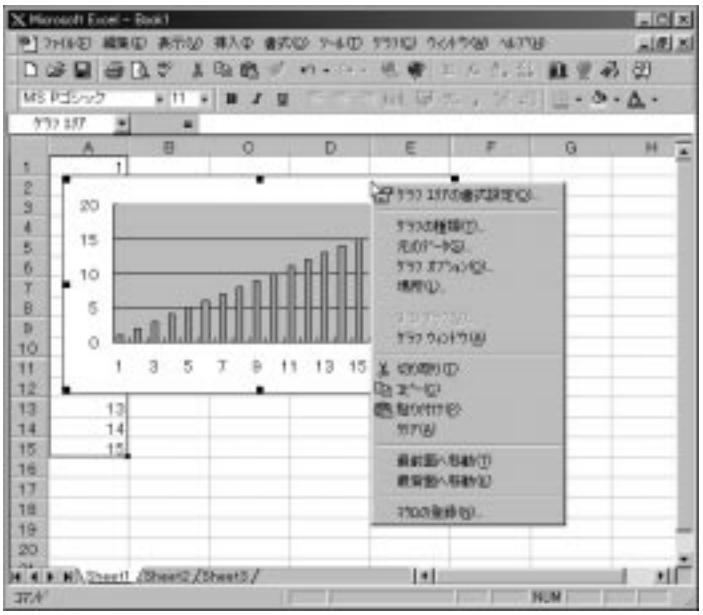

図 6.6: EXCEL で EPS ファイル作成


図 6.7: 仮想プリンタのフォント

### 6.4.2 Micrografx Designer による図の作成

Micrografx Designer を立ち上げます（これについては研究室共有のアプリケーションとして保管してありますので，各自インストールしておいてください）．そこで，実際に図を作ってみます．

そしてこれを eps 形式で保存します．メニューのファイルからプリンタの設定を選んで，Excel でやったようにプリンタ名に仮想プリンタを選び，プロパティの PostScript でカプセルかされた PostScript（EPS 形式）を選び，OK を押します．先ほどのように PS フォントで図を作成したら，図 6.8 のように，メニューの印刷の範囲選択を選んで，自分が作成した図を囲んでください．するとファイル名を聞いてくるので，そこで自分のファイル名に EPS の拡張子をつけましょう．これで Micrografx Designer でも，EPS ファイルを作れることが分かりました．

この他にも Adobe Illustrator 8.0j などさまざまなアプリケーションがありますが，研究室共有のものではないので，各自でインストールしてもらうことになります．

図 6.8: Micrografx Designer で EPS ファイル作成

# 第7章　ファイルの変換

## 7.1　ファイル変換の必要性

前回の LaTeX ゼミでは、秀丸エディタを使って LaTeX のソースを作成し、GUIShell でコンパイルし、DVIOUT を使ってその結果を表示するという一連の作業について説明しました。DVIOUT にも印刷機能がありますから、ここまでで LaTeX の文書を印刷することはできるようになりました。

知的システムデザイン研究室では、研究成果をホームページ上に公開しています。私たちの月例発表会のレジュメや、卒業論文もホームページで公開されます。その際に、TeX のファイルをそのまま公開しても、コンパイルしなければ見ることができないため、公開する意味があまりありません。DVI ファイルの場合も、DVIOUT がインストールされていなければ見ることができないので、あまり現実的な方法ではありません。

そこで、我々の研究室では、論文や月例発表会のレジュメは PDF ファイルに変換して公開しています。TeX のファイルを PDF ファイルに変換するためには図 7.1 のような手順が必要になります。

図 7.1 のファイル形式の下にはその形式を表示または編集するのに必要となるアプリケーションです。また、点線の丸で囲んだソフトは、これからインストールしなければならないアプリケーションです。

## 7.2　DVI→PS

PS ファイルとはポストスクリプトファイルのことです。ポストスクリプトは、Adobe Systems 社が開発したページ記述言語です。高品位の印刷が可能なため、DTP 用のレイアウトソフトがこの形式を採用しています。

DVI ファイルを PS ファイルに変換するためには、前回インストールした GUIShell を利用します。TEX ファイルをコンパイルし、GUIShell の左から 7 番目のボタンをクリックしてください。（図 7.2 を参照）

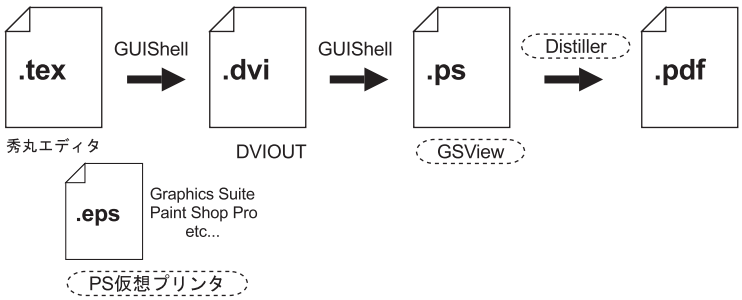


図 7.1: ファイルの変換

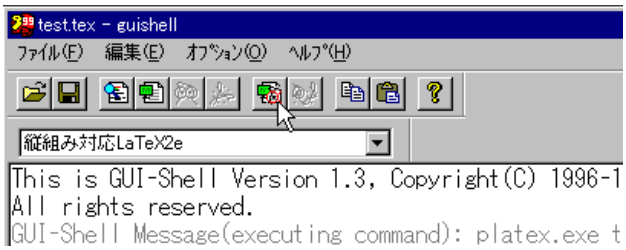


図 7.2: GUIShell で PS ファイルに変換

これで、TEX ファイルがあったフォルダに、PS ファイルが作成されます。例えば test.tex というファイルであれば test.ps というファイルが作られます。ポストスクリプトファイルの名前には拡張子 .ps を付けるのが決まりになっています。

## 7.3 PS ファイルを見るために

Ghostscript/GSView

せっかく、PS ファイルを作成してもこのままでは見ることができません。PS ファイルを見るためには、Ghostscript と GSView というアプリケーションをインストールしなければなりません。

Ghostscript は Postscript のエミュレータで、Postscript 形式（PS）や EPS 形式の画像ファイルや文書ファイルを別の画像形式に変換したりすることができます。また、GSView は、内部的に Ghostscript を動かすことによって、PS ファイルや EPS ファイルを画面上で確認することを可能にしています。

Ghostscript のインストールの詳細については、別途マニュアルを作成していますのでそちらを参照してください。

## 7.4 PS → PDF

**PDF とは**

PDF とは、Adobe Systems 社が開発した一つのファイルフォーマット形式です。この形式の書類はプラットフォームを選ばず、Windows や Macintosh、Unix 上で、共通に取り扱うことが出来[1]、作成元で使用したアプリケーションやフォントが受け取り側のパソコンにインストールされていなくても、その書類のレイアウト情報を保持したまま、閲覧、印刷できるという優れたものです。この共通のファイルフォーマットされた形式を PDF(Portable Document File)Format と呼びます。

もう一つの特徴としては、ファイルサイズが非常にコンパクトに圧縮されるため、ネットワークやインターネットを通じた配信に向いています。この PDF 書類を閲覧・印刷するためのソフトが Adobe Acrobat Reader3.0J でフリーウエアとして、アドビ社より配布されています。勿論、再配布も可能で、作成した PDF ファイルに添付して配布することもできます。

PDF を作成するためには、Adobe Acrobat 3.0 が必要です。Acrobat 3.0 は，表 7.4 に示すアプリケーション及びドライバー類で構成されます。

[1] Acrobat の英語版がサポートするプラットフォームは Macintosh, Windoew3.1, 95, Unix, OS/2 である。日本語版では、Macinsosh, Windows95/NT がサポート出来る。

表 7.1: Acrobat 3.0 の構成

| | |
|---|---|
| Acrobat Distiller 3.0 | ポストスクリプトファイルから PDF ファイルを作成する |
| Acrobat Exchange 3.0 | PDF ファイルブックマークやサムネイルを付けたり、リンクを張ったり、又は、セキュリティの設定などのオーサーリングする |
| PDF Writer | 作成した書類から簡易的に PDF ファイルを作成する一種のプリンタードライバー |
| PSPrinter ドライバー | ポストスクリプトファイルを作成するプリンタードライバー |

図 7.3: Distiller による変換

**PS ファイルを PDF に変換**

PS ファイルを PDF に変換するためには、Acrobat Distiller を利用します。Distiller の使い方は簡単です。インストールした時点で、Distiller が PS ファイルに関連付けされるので、PS ファイルをダブルクリックするだけで、変換作業が行われ（図 7.3）、同じフォルダに PDF ファイルが作成されます。たとえば、test.ps というファイルであれば test.pdf というファイルが作られます。

Distiller をスタートメニューから起動し、［ファイル］－［開く］から PS ファイルを指定するという方法もあります。

Adobe Acrobat のインストール方法は、ソフトに添付のマニュアルを参照してください。マニュアルなしでもウィザードに従っていけば簡単にインストールできると思います。

以上のような手順で、PDF ファイルを作成することができます。作成した PDF ファイルをホームページ上に公開する方法については、ホームページゼミで学んでください。

## 7.5 EPS ファイル

EPS 形式とは、低解像度用と高解像度用の両方のデータを内部に持つ PostScript ファイルの形式です。画面上では低解像度用のデータで高速に編集することができ、印刷時には高解像度用のデータでプリンタの最高性能の出力を得ることができるという利点を持っています。

高品質の印刷結果を得るために、TEX の文書に張り付ける画像は EPS ファイルにしてください。EPS

ファイルは、Paint Shop Pro や Photo Shop、Illustrator などのソフトで作成することができます。BMP ファイルなどを使いたいときは、いちどこれらのソフトに取り込んで、ファイル形式を EPS 形式にして保存してください。

EPS ファイルの拡張子は.eps ですが、単にファイルの拡張子を変えるだけでは、ファイル形式は変わらないので注意してください。

EXCEL 作成したグラフなどを EPS ファイルにする際には仮想 Postscript プリンタを使う方法がありますが、それについては後で述べます。