Computing in the Department of Statistics at Penn State University

The Documentation Project

Contents

Chapter 1

LaTeX

LaTeX is a macro package which enables authors to typeset and print their work with the highest typographical quality, using a predefined, professional layout. Since its introduction, it has been periodically updated and revised, like all software products. For many years now the version number has been fixed at 2ε. It processes an input file containing the text of a document with interspersed commands that describe how the text should be formatted. It produces at least three files as output:

1. A "Device Independent", or '.dvi' file. This contains commands that can be translated into commands for a variety of output devices. You can view the output of LaTeX by using a program such as xdvi, which actually uses the '.dvi' file.

2. A "transcript" or '.log' file that contains summary information and diagnostic messages for any errors discovered in the input file.

3. An "auxiliary" or '.aux' file. This is used by LaTeX itself, for things such as sectioning.

Usage

- Create a text file with extension tex in your favorite text editor. e.g. `mydoc.tex`
- Compile it with latex

```
lightning:~  /temp> latex mydoc.tex
```

- Now you can view the final output by typing

```
lightning:~  /temp> xdvi mydoc
```

or

```
lightning:~  /temp> xdvi mydoc.dvi
```

- To print the output directly to the printer

```
lightning:~  /temp> dvips mydoc -Pprinter_name
```

If you do not give the `-P` option it will print to your default printer.

- To save the output to a postscript file type

```
lightning:~  /temp> dvips mydoc -o mypsfile.ps
```

Again if you do not specify any output file it will be saved in `mydoc.ps`. You can view the postscript file by typing

```
lightning:~  /temp> gv mydoc.ps &
```

1.1 LaTeX Input Files

The input for LaTeX is a plain ASCII text file. You can create it with any text editor. It contains the text of the document as well as the commands which tells LaTeX how to typeset the text. The commands start with a '\' (backslash character).

Eg: `\bf, \it,` etc.

When LaTeX2ϵ process an input file, it requires us to follow a certain structure. Thus every input file must start with the command

```
\documentstyle[options]{style}
```

When all the set up work is done, you start the body of the text with the command

```
\begin{document}
```

Now you enter the text mixed with some useful LaTeX commands. At the end of the document you add the following command

```
\end{document}
```

1.2 Preamble- Style files and Inclusion of packages

Preamble is the portion between `\documentclass` and `\begin{document}`. This can contain package loading command like `\usepackagepackagename`. Any number of `\usepackage` command can be issued or alternatively you can give the package-names as a comma separated list in a single `\usepackage` command.

separated list in a single `\usepackage` command.

Preamble can also contain the header/footer style chosen, the command for which take the following form:

```
\pagestyle{styleoption}
```

The style options available are `empty` (header and footer empty), `plain` (page number in the footer alone, no header), —`headings` (chapter heading in odd header and section heading in even header, no footers), `myheadings` (user defined text in

odd and even headers, no footers). You can also define your own custom headers and footers with fancied text, boxes, graphic elements, etc.

A typical preamble of a LaTeX document will look like:

```
\documentclass[a4paper,11pt,twocolumn]{article}
\usepackage{amsmath,times}
\pagestyle{headings}
\begin{document}
```

1.2.1 Different Document Styles

The style files determines the layout of the final document after compiling with latex. You can thibk of it as templates for different purpose (books, letters, journal articles). The most common LaTeX document styles include:

- article
- report
- letter
- book

They are selected with the following command:

```
\documentstyle[options]{style}
```

The `options` available allow various modifications to be made to the formatting, like selecting font size — `10pt, 11pt, 12pt`, specifying paper size — `letterpaper, legalpaper, executivepaper`, page formats — `onecolumn, twocolumn` etc.

Some of the options for the different styles are:

- article: 11pt, 12pt, twoside, twocolumn, draft, fleqn, leqno, acm
- report: 11pt, 12pt, twoside, twocolumn, draft, fleqn, leqno, acm
- letter: 11pt, 12pt, fleqn, leqno, acm
- book: 11pt, 12pt, twoside,twocolumn, draft, fleqn, leqno

If you specify more than one option, they must be separated by a comma. For example

```
\documentstyle[12pt,twocolumn]{article}
```

Note that you can design your own style file for a specific purpose (e.g. Thesis style, Poster style, Seminar Style). In this section we will discuss some commonly used style files.

1.2.2 Inclusion of Packages

Packages for formatting can be included to enhance the appearance of the latex document. The general format for including packages is

```
\usepackage[options]{package_name}
```

e.g

```
\usepackage[usenames]{color}
```

detailed discussion about some useful packages are given in section 1.10.

1.3 Bibliography

Bibliography is the environment, which helps the author to cross-reference one publication from the list of sources at the end of the document. Bibliography needs consistency, LaTeX helps author to write well structured bibliography, because this is how LaTeX works—by specifying structure.

It is easy to convert the style of bibliography to publisher's require, without touching the code inside the bibliography. We can maintain a bibliographic data base using the program `\BibTeX` . While preparing the articles, we can extract the needed references in needed style from this data base. Harvard and natbib are widely used packages for generating bibliography.

To produce bibliography, we have the environment `thebibliography`[1], which acts similar to `enumerate` environment.

INPUT

[1]Bibiliography environment need two compilation. In first compilation it will generate file with aux extension, where `citation` and `bibcite` will be marked and in second compilation \cite will be replaced by numeral or author-year code.

```
It is  hard to write unstructured and disorganised documents using
\LaTeX~\cite{les85}.It is interesting to type set one
equation~\cite[Sec 3.3]{les85} rather than setting 10 pages of
running matter~\cite{don89,rondon89}.

\begin{thebibliography}{9}
\bibitem{les85}Leslie Lamport, 1985. \emph{\LaTeX---A Document
Preparation System---User's Guide and Reference Manual},
Addision-Wesley, Reading.

\bibitem{don89}Donald E. Knuth, 1989. \emph{Typesetting Concrete
Mathematics}, TUGBoat, 10(1):31-36.

\bibitem{rondon89}Ronald L. Graham, Donald E. Knuth, and Ore
Patashnik, 1989. \emph{Concrete Mathematics: A Foundation for
Computer Science}, Addison-Wesley, Reading.
\end{thebibliography}
```

OUTPUT

It is hard to write unstructured and disorganized documents using LaTeX [1]. It is interesting to type set one equation [1, Sec 3.3] rather than setting 10 pages of running matter [2,3].

Bibliography

[1] Leslie Lamport, 1985. *LaTeX—A Document Preparation System—User's Guide and Reference Manual*, Addision-Wesley, Reading.

[2] Donald E. Knuth, 1989. *Typesetting Concrete Mathematics*, TUGBoat, 10(1):31-36.

[3] Ronald L. Graham, Donald E. Knuth, and Ore Patashnik, 1989. *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley, Reading.

1.4 Mathematical formulae

LaTeX has a special mode for typesetting mathematics. Mathematical text within a paragraph is entered between `\(` and `\)`, between `$` and `$` or between `\begin{math}` and `\end{math}`.

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach:
$c^{2}=a^{2}+b^{2}$
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach: $c^2 = a^2 + b^2$

```
\TeX{} is pronounced as
$\tau\epsilon\chi$.\\[6pt]
100~m$^{3}$ of water\\[6pt]
This comes from my $\heartsuit$
```

TEX is pronounced as $\tau\epsilon\chi$.

100 m^3 of water

This comes from my $\heartsuit$

It is preferable to *display* larger mathematical equations or formulae, rather than to typeset them on separate lines. This means you enclose them in `\[` and `\]` or between `\begin{displaymath}` and `\end{displaymath}`. This produces formulae which are not numbered. If you want LaTeX to number them, you can use the `equation` environment.

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach:
\begin{displaymath}
c^{2}=a^{2}+b^{2}
\end{displaymath}
And just one more line.
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach:

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

And just one more line.

You can reference an equation with `\label` and `\ref`

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
From (\ref{eq:eps}), we gather
\ldots
```

(1.1) $$\epsilon > 0$$

From (1.1), we gather ...

Note that expressions will be typeset in a different style if displayed:

```
$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$
```

$\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$$

1. Most spaces and linebreaks do not have any significance, as all spaces either are derived logically from the mathematical expressions or have to be specified using special commands such as `\,`, `\quad` or `\qquad`.
2. Empty lines are not allowed. Only one paragraph per formula.
3. Each letter is considered to be the name of a variable and will be typeset as such. If you want to typeset normal text within a formula (normal upright font and normal spacing) then you have to enter the text using the `\textrm{...}` commands.

```
\begin{equation}
\forall x \in \mathbf{R}:
\qquad x^{2} \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R}: \qquad x^2 \geq 0 \tag{1.2}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\textrm{for all }x\in\mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \qquad \text{for all } x \in \mathbf{R} \tag{1.3}$$

Mathematicians can be very fussy about which symbols are used: it would be conventional here to use 'blackboard bold', which is obtained using `\mathbb` from the package amsfonts or amssymb. The last example becomes

```
\begin{displaymath}
x^{2} \geq 0\qquad
\textrm{for all }x\in\mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \qquad \text{for all } x \in \mathbb{R}$$

1.4.1 Grouping in Math Mode

Most math mode commands act only on the next character. So if you want a command to affect several characters, you have to group them together using curly braces: `{...}`.

```
\begin{equation}
a^x+y \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \tag{1.4}$$

1.4.2 Building Blocks of a Mathematical Formula

Now the most important commands used in mathematical typesetting will be described.

Lowercase Greek letters are entered as `\alpha`, `\beta`, `\gamma`, ..., uppercase letters are entered as `\Gamma`, `\Delta`, ...[2]

```
$\lambda,\xi,\pi,\mu,\Phi,\Omega$
```

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

Exponents and Subscripts can be specified using the `^` and the `_` character.

```
$a_{1}$ \qquad $x^{2}$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$\\
$e^{x^2} \neq {e^x}^2$
```

$a_1 \qquad x^2 \qquad e^{-\alpha t} \qquad a^3_{ij}$
$e^{x^2} \neq e^{x2}$

The **square root** is entered as `\sqrt`, the n^{th} root is generated with `\sqrt[`n`]`. The size of the root sign is determined automatically by LaTeX. If just the sign is needed, use `\surd`.

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$\\[3pt]
$\surd[x^2 + y^2]$
```

$\sqrt{x} \qquad \sqrt{x^2+\sqrt{y}} \qquad \sqrt[3]{2}$
$\surd[x^2 + y^2]$

The commands `\overline` and `\underline` create **horizontal lines** directly over or under an expression.

```
$\overline{m+n}$
```

$\overline{m+n}$

The commands `\overbrace` and `\underbrace` create long **horizontal braces** over or under an expression.

[2] There is no uppercase Alpha defined in LaTeX 2_ε because it looks the same as a normal roman A. Once the new math coding is done, things will change.

```
$\underbrace{ a+b+\cdots+z }_{26}$
```

$$\underbrace{a+b+\cdots+z}_{26}$$

To add mathematical accents such as small arrows or tilde signs to variables, you can use the commands given in Table ?? on page ??. Wide hats and tildes covering several characters are generated with `\widetilde` and `\widehat`. The `'` symbol gives a prime.

```
\begin{displaymath}
y=x^{2}\qquad y'=2x\qquad y''=2
\end{displaymath}
```

$$y = x^2 \qquad y' = 2x \qquad y'' = 2$$

Vectors often are specified by adding small arrow symbols on top of a variable. This is done with the `\vec` command. The two commands `\overrightarrow` and `\overleftarrow` are useful to denote the vector from A to B.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

$$\vec{a} \quad \overrightarrow{AB}$$

Names of log-like functions are often typeset in an upright font and not in italic like variables. Therefore LaTeX supplies the following commands to typeset the most important function names:

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

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

For the modulo function, there are two commands: `\bmod` for the binary operator "a mod b" and `\pmod` for expressions such as "$x \equiv a \pmod{b}$."

A built-up **fraction** is typeset with the `\frac{...}{...}` command. Often the slashed form 1/2 is preferable, because it looks better for small amounts of 'fraction material.'

```
$1\frac{1}{2}$~hours
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

$1\frac{1}{2}$ hours

$$\frac{x^2}{k+1} \qquad x^{\frac{2}{k+1}} \qquad x^{1/2}$$

To typeset binomial coefficients or similar structures, you can use either the command `{... \choose ...}` or `{... \atop ...}`. The second command produces the same output as the first one, but without braces. (Note that the usage of these old-style commands is expressly forbidden by the amsmath package. They are replaced by `\binom` and `\genfrac`. The latter is a superset of all related constructs. e.g. you may get a similar construct to `\atop` by `\newcommand{\newatop}[2]{\genfrac{}{}{0pt}{1`

```
\begin{displaymath}
{n \choose k}\qquad {x \atop y+2}
\end{displaymath}
```

$$\binom{n}{k} \qquad \begin{matrix} x \\ y+2 \end{matrix}$$

For binary relations it may be useful to stack symbols over each other. `\stackrel` puts the symbol given in the first argument in superscript-like size over the second which is set in its usual position.

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

The **integral operator** is generated with `\int`, the **sum operator** with `\sum` and the **product operator** with `\prod`. The upper and lower limits are specified with `^` and `_` like subscripts and superscripts. [3]

```
\begin{displaymath}
\sum_{i=1}^{n} \qquad
\int_{0}^{\frac{\pi}{2}} \qquad
\prod_\epsilon
\end{displaymath}
```

$$\sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad \prod_{\epsilon}$$

For **braces** and other delimiters, there exist all types of symbols in TEX (e.g. $[\ \langle\ \|\ \updownarrow$). Round and square braces can be entered with the corresponding keys, curly braces with `\{`, all other delimiters are generated with special commands (e.g. `\updownarrow`). For a list of all delimiters available, check table **??** on page **??**.

[3] AMS-LATEXin addition has multiline super-/subscripts

```
\begin{displaymath}
{a,b,c}\neq\{a,b,c\}
\end{displaymath}
```

$$a,b,c \neq \{a,b,c\}$$

If you put the command `\left` in front of an opening delimiter or `\right` in front of a closing delimiter, TeX will automatically determine the correct size of the delimiter. Note that you must close every `\left` with a corresponding `\right`, and that the size is determined correctly only if both are typeset on the same line. If you don't want anything on the right, use the invisible '`\right.`'!

```
\begin{displaymath}
1 + \left( \frac{1}{ 1-x^{2} }
    \right) ^3
\end{displaymath}
```

$$1 + \left(\frac{1}{ 1-x^{2} } \right) ^3$$

In some cases it is necessary to specify the correct size of a mathematical delimiter by hand, which can be done using the commands `\big`, `\Big`, `\bigg` and `\Bigg` as prefixes to most delimiter commands.[4]

```
$\Big( (x+1) (x-1) \Big) ^{2}$\\
$\big(\Big(\bigg(\Bigg($\quad
$\big\}\Big\}\bigg\}\Bigg\}$\quad
$\big\|\Big\|\bigg\|\Bigg\|$
```

$$\Big((x+1)(x-1)\Big)^2$$
$$\big(\Big(\bigg(\Bigg(\quad \Big\}\bigg\}\Big\} \quad \big\|\Big\|\bigg\|\Bigg\|$$

To enter **three dots** into a formula, you can use several commands. `\ldots` typesets the dots on the baseline, `\cdots` sets them centred. Besides that, there are the commands `\vdots` for vertical and `\ddots` for diagonal dots. You can find another example in section 1.4.4.

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

$$x_1,\ldots,x_n \qquad x_1+\cdots+x_n$$

1.4.3 Math Spacing

If the spaces within formulae chosen by TeX are not satisfactory, they can be adjusted by inserting special spacing commands. There are some commands for small

[4] These commands do not work as expected if a size changing command has been used, or the `11pt` or `12pt` option has been specified. Use the exscale or amsmath packages to correct this behaviour.

spaces: `\,` for $\frac{3}{18}$ quad (␣), `\:` for $\frac{4}{18}$ quad (␣) and `\;` for $\frac{5}{18}$ quad (␣). The escaped space character `\␣` generates a medium sized space and `\quad` (␣) and `\qquad` (␣) produce large spaces. The size of a `\quad` corresponds to the width of the character 'M' of the current font. The `\!` command produces a negative space of $-\frac{3}{18}$ quad (␣).

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_{D} g(x,y)
  \, \ud x\, \ud y
\end{displaymath}
instead of
\begin{displaymath}
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}
```

$$\iint_D g(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

instead of

$$\int\int_D g(x,y)\mathrm{d}x\mathrm{d}y$$

Note that 'd' in the differential is conventionally set in roman.

$\mathcal{AMS}$-LaTeX provides another way for finetuning the spacing between multiple integral signs, namely the `\iint`, `\iiint`, `\iiiint`, and `\idotsint` commands. With the amsmath package loaded, the above example can be typeset this way:

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \, \ud x \, \ud y
\end{displaymath}
```

$$\iint_D \mathrm{d}x\,\mathrm{d}y$$

See the electronic document testmath.tex (distributed with $\mathcal{AMS}$-LaTeX) or Chapter 8 of "The LaTeX Companion"[5] for further details.

1.4.4 Vertically Aligned Material

To typeset **arrays**, use the `array` environment. It works somewhat similar to the `tabular` environment. The `\\` command is used to break the lines.

[5] available at `CTAN:/tex-archive/info/ch8.*`.

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

The `array` environment can also be used to typeset expressions which have one big delimiter by using a "`.`" as an invisible `\right` delimiter:

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 a & \textrm{if $d>c$}\\
 b+x & \textrm{in the morning}\\
 l & \textrm{all day long}
  \end{array} \right.
\end{displaymath}
```

$$y = \left\{ \begin{array}{ll} a & \textrm{if } d > c \\ b+x & \textrm{in the morning} \\ l & \textrm{all day long} \end{array} \right.$$

As within the `tabular` environment you can also draw lines in the `array` environent, e.g. separating the entries of a matrix:

```
\begin{displaymath}
\left(\begin{array}{c|c}
 1 & 2 \\
\hline
3 & 4
\end{array}\right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}\right)$$

For formulae running over several lines or for equation systems, you can use the environments `eqnarray`, and `eqnarray*` instead of `equation`. In `eqnarray` each line gets an equation number. The `eqnarray*` does not number anything.

The `eqnarray` and the `eqnarray*` environments work like a 3-column table of the form `{rcl}`, where the middle column can be used for the equal sign or the not-equal sign. Or any other sign you see fit. The `\\` command breaks the lines.

```
\begin{eqnarray}
f(x) & = & \cos x     \\
f'(x) & = & -\sin x   \\
\int_{0}^{x} f(y)dy &
 = & \sin x
\end{eqnarray}
```

$$\begin{eqnarray} f(x) &=& \cos x \tag{1.5} \\ f'(x) &=& -\sin x \tag{1.6} \\ \int_0^x f(y)dy &=& \sin x \tag{1.7} \end{eqnarray}$$

Notice that the space on either side of the the equal signs is rather large. It can be reduced by setting `\setlength\arraycolsep{2pt}`, as in the next example.

Long equations will not be automatically divided into neat bits. The author has to specify where to break them and how much to indent. The following two methods are the most common ones used to achieve this.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x -\frac{x^{3}}{3!}
     +\frac{x^{5}}{5!}-{}
                      \nonumber\\
 & & {}-\frac{x^{7}}{7!}+{}\cdots
\end{eqnarray}}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \tag{1.8}$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1
     -\frac{x^{2}}{2!} +{} }
                      \nonumber\\
 & & {}+\frac{x^{4}}{4!}
     -\frac{x^{6}}{6!}+{}\cdots
\end{eqnarray}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \tag{1.9}$$

The `\nonumber` command causes LaTeX to not generate a number for this equation.

It can difficult be to get vertically aligned equations to look right with these methods; the package amsmath provides a more powerful set of alternatives. (see `split` and `align` environments).

1.4.5 Theorems, Laws, ...

When writing mathematical documents, you probably need a way to typeset "Lemmas", "Definitions", "Axioms" and similar structures. LaTeX supports this with the command

`\newtheorem{`*name*`}[`*counter*`]{`*text*`}[`*section*`]`

The *name* argument, is a short keyword used to identify the "theorem". With the *text* argument, you define the actual name of the "theorem" which will be printed in the final document.

The arguments in square brackets are optional. They are both used to specify the numbering used on the "theorem". With the *counter* argument you can specify the *name* of a previously declared "theorem". The new "theorem" will then be

numbered in the same sequence. The *section* argument allows you to specify the sectional unit within which you want your "theorem" to be numbered.

After executing the `\newtheorem` command in the preamble of your document, you can use the following command within the document.

`\begin{`*name*`}[`*text*`]`
This is my interesting theorem
`\end{`*name*`}`

This should be enough theory. The following examples will hopefully remove the final remains of doubt and make it clear that the `\newtheorem` environment is way too complex to understand.

```
% definitions for the document
% preamble
\newtheorem{law}{Law}
\newtheorem{jury}[law]{Jury}
%in the document
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}\end{jury}
\begin{law}No, No, No\end{law}
```

Law 1 *Don't hide in the witness box*

Jury 2 (The Twelve) *It could be you! So beware and see law 1*

Law 3 *No, No, No*

The "Jury" theorem uses the same counter as the "Law" theorem. Therefore it gets a number which is in sequence with the other "Laws". The argument in square brackets is used to specify a title or something similar for the theorem.

```
\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
If there are two or more
ways to do something, and
one of those ways can result
in a catastrophe, then
someone will do it.\end{mur}
```

Murphy 1.4.1 *If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.*

The "Murphy" theorem gets a number which is linked to the number of the current section. You could also use another unit, for example chapter or subsection.

1.5 Graphics and Figure environment

The easiest way to include graphics (postscript) figures in your document is

```
\includegraphics{yourfile.ps}
```

You should include the "grpahics" package in the preamble

```
\usepackage{graphics}
```

To scale, use some optional arguments

```
\includegraphics[width=5cm,height=10cm]{yourfile.ps}
```

would rescale the postscript so that it was 5cm wide and 10cm high. To make the picture 5cm wide and scale the height in proportion use

```
\includegraphics[width=5cm]{yourfile.ps}
```

To rotate anticlockwise by the specified number of degrees, use

```
\includegraphics[angle=150]{yourfile.ps}
```

To trim the figure you can specify the amount to remove from the left, bottom, right and top side (e.g. trim=1 2 3 4)

```
\includegraphics[ trim=1 2 3 4]{yourfile.ps}
```

Other packages for inclusion of postscript figures exits.(e.g psfig).

Whatever graphics you want to add, you should use the figure environment so that LATEX can cope sensibly with situations where, for example, you attempt to insert near the bottom of a page a graphic that's half a page high. The figure environment will float the graphic to the top or bottom of the page, or on the next page, with preferences that you can provide.

`h` here
`t` top of page
`b` bottom of page
`p` on a page with no text

Putting ! as the first argument in the square brackets will encourage LATEX to do what you say, even if the result's sub-optimal. See the online hints about floats in LaTeX for further details.

```
\begin{figure}[htbp]
   \vspace{0.5in}
   \caption{0.5 inch of space}
\end{figure}
%
```

Figure 1.1: 0.5 inch of space

Similarly

```
\begin{figure}[hbtp]
  \includegraphics[height=40mm]{yourfile.ps}
  \includegraphics[angle=120, height=20mm]{yourfile.ps}
\caption{Tigers}
\end{figure}
```

1.6 Tables

The table environment can be best described by some short examples. For example the following input file

```
\begin{tabular}{rlcrr}  % column alignment r=right l=left c=center
 Number & Team & Attempts & Won & Lost \\
 1 & Wolverines   & 349 & 233 & 116 \\
 2 & Haberdashers & 348 & 200 & 148 \\
 3 & Coopers      & 349 & 175 & 174 \\
 4 & Debuggers    & 340 &  99 & 241 \\
 5 & Pirates      & 339 &   8 & 331
\end{tabular}
```

produces

```
\begin{tabular}{r|l||c||rr|}
 Number & Team & Attempts & Won & Lost \\
 1 & Wolverines   & 349 & 233 & 116 \\
 2 & Haberdashers & 348 & 200 & 148 \\
```

Table 1.1: Example Table 1

Number	Team	Attempts	Won	Lost
1	Wolverines	349	233	116
2	Haberdashers	348	200	148
3	Coopers	349	175	174
4	Debuggers	340	99	241
5	Pirates	339	8	331

```
 3 & Coopers      & 349 & 175 & 174 \\
 4 & Debuggers    & 340 &  99 & 241 \\
 5 & Pirates      & 339 &   8 & 331
\end{tabular}
```

Table 1.2: Example Table 2

Number	Team	Attempts	Won	Lost
1	Wolverines	349	233	116
2	Haberdashers	348	200	148
3	Coopers	349	175	174
4	Debuggers	340	99	241
5	Pirates	339	8	331

```
\begin{tabular}{|r|l||c||rr|}
        \hline
 Number & Team & Attempts & Won & Lost \\
        \hline \hline
 1 & Wolverines   & 349 & 233 & 116 \\
        \hline
 2 & Haberdashers & 348 & 200 & 148 \\
        \hline
 3 & Coopers      & 349 & 175 & 174 \\
        \hline
 4 & Debuggers    & 340 &  99 & 241 \\
        \hline
 5 & Pirates      & 339 &   8 & 331 \\
        \hline
\end{tabular}
```

Table 1.3: Example Table 2

Number	Team	Attempts	Won	Lost
1	Wolverines	349	233	116
2	Haberdashers	348	200	148
3	Coopers	349	175	174
4	Debuggers	340	99	241
5	Pirates	339	8	331

1.7 Cross Referencing

One of the strengths of LaTeX is its capability to cross-reference through information it places in auxilary files. Done properly, this feature permits one to extract, insert, move and modify large and small chunks of the document around without having to manually renumber cross-references, it is all done automatically.

1.7.1 Table of contents

One of the easiest things to do is to insert a table of contents into a document simply by placing the LaTeX command

```
\tableofcontents
```

at the desired location in the document

The first time this is LaTeXed the table will not appear because the information is being stored in the associated .toc file. Subsequent LaTeXing will typeset the table of contents.

1.7.2 Footnotes

Unlike most publishers, LaTeX easily handles footnotes with comtemptuous ease. Just use the

```
\footnote{some-text}
```

command with one argument being the text of the footnote.This will typeset a numerical flag at the location of the footnote command and will place the footnote text at the bottom of the page.

1.7.3 Labels and references

Somewhat more sophisticated are references to equations and sections. First one has to label them as in

```
\section{...}
\label{sec-name}
```

or in

```
\begin{equation}
...
\label{eq-name}
\end{equation}
```

which associates a string such as "sec-name" with the number of the section, and a string such as "eq-name" with the number of the equation. See fractals23.tex.

Having created the labels, you refer to the objects using the `\reflabel-name`.You need to put parentheses around the equation number in its reference.

One also labels and refers to chapters, subsections, subsubsections, tables, figures, and enumerated lists.

Often you begin to lose track of the labels used. In such a case in the preamble will cause names of labels to also appear in a (draft) printed document for your ready reference.

1.8 Playing with Fonts

We give a brief description of the new commands for font selection scheme used in LaTeX2$_e$. A more detailed description with examples is given in The LaTeX Companion, and the interface for class- and package-writers is described in LaTeX2$_e$ Font Selection.

A font is described by three attributes the family (Times Roman, Sans Serif, Typewriter), the series (bold or medium) and the shape (upright, italics, slanting, small caps.

```
\rmfamily
\sffamily
\ttfamily
\mdseries
\bfseries
\upshape
\itshape
\slshape
\scshape
```

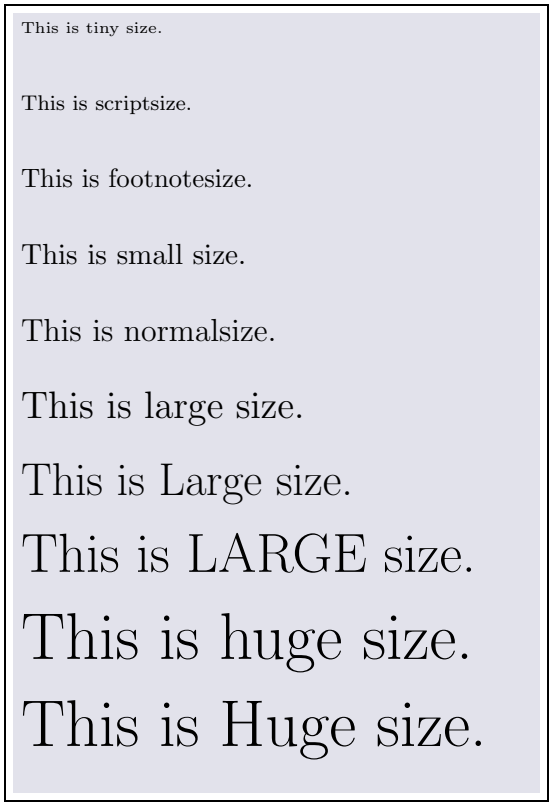
Each command changes just one attribute of the font (the attribute changed is part of the name). One result of this is that, for example, `\bfseries \itshape` produces both a change of series and a change of shape, to give a bold italic font.

```
{\bfseries \itshape bold italic font}
```

produces
bold italic font

The size of the font is determined by the size attribute (footnote, tiny, Huge etc)

```
{\tiny This is tiny size.}

{\scriptsize This is scriptsize.}

{\footnotesize This is footnotesize.}

{\small This is small size.}

{\normalsize This is normalsize.}

{\large This is large size.}

{\Large This is Large size.}

{\LARGE This is LARGE size.}

{\huge This is huge size.}

{\Huge This is Huge size.}
```



Combining the two

```
{\bfseries \itshape\Large Large bold italic font}
```

produces
Large bold italic font

1.9 PSU Thesis Style Files

This one of the most important style files we would need while writing a thesis A zipped archive of the thesis style file can be found at http://www.stat.psu.edu/ surajit/latex/phd.zip

1.10 Useful Packages Available

1.10.1 Color

The easiest way to use color text in LaTeX 2_ε is to use the color package. You load the color package in your preamble as usual

```
\documentclass[11pt]{article}
\usepackage[usenames]{color}
```

Here the option [usenames] causes the definitions for the 68 colors known to dvips (see next page) to be preloaded. Those colors can then be specified by name. For example `{\color{BrickRed} BrickRed}` produces BrickRed. Alternately one can use the command

```
\textcolor{BrickRed}{this is BrickRed}
```

which produces this is BrickRed.

If the 68 predefined colors do not suffice, or if you do not want to load the definitions of all of them, you may define your own custom colors.

Here is a dark blue defined in terms of the RGB (red, green, blue) color model.

```
\definecolor{MyDarkBlue}{rgb}{0,0.08,0.45}
\begin{center}
{\color{MyDarkBlue}This color is MyDarkBlue}
\end{center}
```

This color is MyDarkBlue

Once we have defined the new color we can use it anywhere in the current document. For example, `\textcolor{MyDarkBlue}{a nice blue}` produces a nice blue.

Custom colors may also be defined in terms of the CMYK (cyan, magenta, yellow, black) color model. Here is a light magenta.

```
\definecolor{MyLightMagenta}{cmyk}{0.1,0.8,0,0.1}
\begin{center}
{\color{MyLightMagenta}This color is MyLightMagenta}
\end{center}
```

This color is MyLightMagenta

Table 1.4: **The 68 standard colors known to dvips**

Apricot	Aquamarine	Bittersweet	Black
Blue	BlueGreen	BlueViolet	BrickRed
Brown	BurntOrange	CadetBlue	CarnationPink
Cerulean	CornflowerBlue	Cyan	Dandelion
DarkOrchid	Emerald	ForestGreen	Fuchsia
Goldenrod	Gray	Green	GreenYellow
JungleGreen	Lavender	LimeGreen	Magenta
Mahogany	Maroon	Melon	MidnightBlue
Mulberry	NavyBlue	OliveGreen	Orange
OrangeRed	Orchid	Peach	Periwinkle
PineGreen	Plum	ProcessBlue	Purple
RawSienna	Red	RedOrange	RedViolet
Rhodamine	RoyalBlue	RoyalPurple	RubineRed
Salmon	SeaGreen	Sepia	SkyBlue
SpringGreen	Tan	TealBlue	Thistle
Turquoise	Violet	VioletRed	White
WildStrawberry	Yellow	YellowGreen	YellowOrange

1.10.2 Other packages

Other packages which can be used to enhance the latex document are

- PsTrick
- Fancy Box
- Fancy header
- Color Table

. Many more packages are listed in http://tex.loria.fr/english/packages.html

Bibliography

[1] H. Kopka and P.W. Daly. **A Guide to LaTeX, 3rd Edition** Addison-Wesley, California 1999.

[2] L. Lamport. **LaTeX A Document Preparation System** Addison-Wesley, California 1986.