1 LaTeX-FAQ

What is LaTeX? LaTeX is a formal language for document type setting, similar to HTML for Web-page design.

Why not MS Word? A short list:

- LaTeX helps adhering to typographical standards (Word is typewriting, LaTeX is typesetting).
- Better support for mathematics (formulae, tables).
- Global bibliographies possible (across documents, machines, authors).
- LaTeX doesn't crash, practically bug-free.
- Document format compatible across platforms and stable over time (only two versions in the past 10 years).
- Much support for automation and templates.
- Numbering (footnotes, figures, sections) works properly.
- Extensible by user, many packages available on the Web.
- LaTeX is free and available for all platforms.
- You can use your favorite text-editor to write LaTeX-documents.
- LaTeX has a cool logo.
- Supported by CS-GSO `;-)`

When to use and *not* to use it. LaTeX is not for everyone—too complicated for average user. Recommended for Mathematics, Physics, Computer Science. 80% of CS research papers written in LaTeX. Good for writing letters; presentations also possible, but one needs to *program* documents. Compromise: *LyX*, but currently only under Linux (WYSIWYM-Concept: What You See Is What You Mean).

I don't like it, it's so complicated. Well, initially, LaTeX is more complicated than *Corel WordPerfect* or *Lotus WordPro* or *Microsoft Word* or *StarOffice* because you have to remember commands. But the result is worth it. Documents look good. For questions:

- Books. Especially: Helmut Kopka, Patrick W. Daly: *A Guide to Latex2ε : Document Preparation for Beginners and Advanced Users*
- `http://www.cs.WPI.EDU/~csgso/latex.html` for a start
- `http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/` for general info on everything
- American Mathematical Society at `http://www.ams.org/tex`
- The TeX User Group at `http://www.tug.org`

What about LaTeX on a Windows-PC? Download LaTeX for Windows (MIKTeX), emacs for Windows, and/or Ghostview for Windows from `ftp://sky.wpi.edu/shareware`.

2 Get started with LaTeX

2.1 A Sample Document (Source Code).

```
\documentclass[11pt]{article}     % YOUR INPUT FILE MUST CONTAIN THESE
\begin{document}                  % TWO LINES PLUS THE \end COMMAND AT
                                  % THE END

\section{Simple Text}          % THIS COMMAND MAKES A SECTION TITLE.

Words are separated by one or    more      spaces.  Paragraphs are
    separated by one or more blank lines.  The output is not affected
by adding extra spaces or extra blank lines to the input file.

Double quotes are typed like this: ``quoted text''.
Single quotes are typed like this: `single-quoted text'.

Dashes between words are typed as three dash characters---like this.

Italic text is typed like this: \emph{this is italic text}.
Bold   text is typed like this: \textbf{this is  bold  text}.

\subsection{A Warning.}        % THIS MAKES A SUBSECTION TITLE.

% Comments are preceded by a Percent-Sign (%)
% There are some special characters:  &   $   #   %   _   {   }   ^   ~   \
% which cannot be typed in directly (you can type some preceded by a "\").

   \subsection{Maths.}
   Formula typesetting is really easy:
   \begin{equation} \sum_{i=1}^{n}=\frac{n(n-1)}{2} \end{equation}

\end{document}                    % THE INPUT FILE ENDS LIKE THIS
```

2.2 From the Source Code to the Printer—3 Easy Steps

1. `latex mydocument.tex`
2. `dvips mydocument`
3. `lpr -Pps3 mydocument.ps`

`latex mydocument.tex` generates a number of files, especially `mydocument.dvi`. This can be viewed using `xdvi mydocument`. Converting to PostScript using `dvips` is only necessary once you're ready to print. Note that if you have figures, tables of content, cross-references, bibliographies, you may have to run `latex` up to three times in a row (to get all cross-references right).

2.3 LaTeX for Programmers—How it works.

1. Every document has two parts: a *preamble* (before the `\begin{document}`) and *main text.*

2. Regular text is typed normally, formulae in *math mode.* Math mode is introduced by enclosing the formula in one of three pairs of operators:

   - `$ x $` for formulae in the current line of text.
   - `\[ x \]` for formulae centered in their own paragraph.
   - `\begin{equation} x \end{equation}` as before, but with a equation number at the right margin.

3. Commands start with a \ and can have zero or one optional parameter enclosed in [] as well as zero to many required parameters enclosed in {}. The first command of each document is `\documentclass[option1,option2,...]{classname}` which determines the *class* of the document and takes an optional list of settings such as paper size, orientation, font size, etc. A typical command changes to emphasized font (italics) and is used like this: `Normal text. \emph{Italicized text}. More normal text.`

4. A number of internal *counters* and *lengths* is used to fine-tune LaTeX layout. Those can be overwritten. A typical counter determines the depth to which section headings are numbered and is reset by `\setcounter{secnumdepth}{`*num*`}`. A typical length is the width of the text on the page which is set using `\setlength{\textwidth}{`*length*`}`. Such lengths can be specified in a number of units; some examples are `3.5cm; .4in; 12pt; 1em; 2ex`. *pt* (Point) is the typographical unit of length, in which font sizes are measured, *em* is the *width* of the letter M in the current font (so it changes when you change the document font size!) and *ex* is the *height* of the letter x.

5. The best way of writing LaTeX-documents is to use your favorite programming language editor. In particular, Emacs has a LaTeX-font-lock-mode (switch on using the Emacs-command `ALT-X font-lock-mode`) which colors your source code and makes writing documents much easier. I have seen one person who wrote LaTeX source code in *Word* because of the immediate syntax checking `;-)`.

6. **Customization**: LaTeX is endlessly customizable, by using the included *packages* (documentation from `http://www.math.uncc.edu/tex/texstuff.html`) or through your own macros defined with the `\newcommand` command.

3 A one-page LaTeX Crash Course

Document templates are provided on the CS-GSO homepage for articles, letters, and reports (e.g., theses).

3.1 A Subsection (`\subsection`)

The spacing is done automatically for you, you should never need to change it. This is one major advantage over Word (with all the extra empty lines that you normally type...). If you **have to**: `\vspace{space}, \bigskip, \medskip, \smallskip`

3.1.1 Subsubsection (`\subsubsection`)

`\label{mysection}`
Subsubsection headings are still on a separate line.

Paragraph (`\paragraph`) This is text in the paragraph.

Subparagraph (`\subparagraph`) This is the sub-paragraph text. Its heading is indented further that the others. For reports and books, a command `\chapter` is available for the top level. By the way, we are in Subsubsection 3.1.1 (`\ref{mysection}`) on page 4 (`\pageref{mysection}`).

3.2 Hyphenation and Line Breaking

LaTeX tries to layout paragraphs according to typographical standards. Those are very strict rules—they sometimes fail. Check for "`Underfull \hbox`"-messages in `latex`-output. Fix by rearranging words or use `\begin{sloppypar} ... \end{sloppypar}`.

3.3 Font Styles

`{\small small and \textbf{bold and \emph{emphasized} and back}} and {\Large Large}.`
small and **bold and *emphasized* and back** and Large.

3.4 Lists etc.

- `\begin{itemize} \item... \item... \end{itemize}` is this list.
- `\begin{enumerate} \item... \item... \end{enumerate}` produces a numbered list.
- Lists can be nested.

3.5 Formulae

```
\[\sigma=\sqrt{\frac{1}{n}\sum_{i-1}^{r}(p_i (x_i-x)^2)}\qquad
 \sigma-\textrm{standard deviation}\]
```

$$\sigma=\sqrt{\frac{1}{n}\sum_{i-1}^{r}(p_i (x_i-x)^2)}\qquad \sigma-\textrm{standard deviation}$$

4 LaTeX for Advanced Users—Tips and Tricks

Overview:

- Including Figures
- Minipages—How to place two figures next to each other.
- Tables
- Bibliographies
- Some packages
- Other tips?

4.1 How to Include Graphics Into LaTeX-Documents?

LaTeX is able to include several kinds of graphics formats, but most of them are not very useful (like LaTeX's `picture`-environment, which is a simple programming language for drawing). The reason for that is that LaTeX is older than most graphics formats (including `GIF` and PostScript). Therefore, the main way of including any type of graphics is through PostScript. Formats supported are `eps` and single-page, bounded `ps`. No format that LaTeX can include is supported by any Microsoft software, therefore the conversion from MS-products to LaTeX is difficult. Conversion of bitmaps is easier. Under UNIX, there are several tools that generate `eps` or `ps` directly (`gnuplot` for graphs, `xfig` and `idraw` for general drawings. This section describes how such `eps`-files are included into LaTeX-documents.

To include graphics, you have to use a certain LaTeX package. Thus, the following command has to be included immediately after the `\documentclass{}`-declaration (for LaTeX 2_ε):

```
\usepackage[dvips]{graphicx}
```

It is useful to collect all pictures in a sub-directory of the document directory, for instance called `figures`. If you then want LaTeX to find your figures, you have to define a path for the graphics, with the following command:

```
\graphicspath{{figures/}}
```

This command has to be given before the `\begin{document}` statement. Note the double braces and the `/` at the end of the path. The inclusion of a picture then works as follows (if the `eps`- or `ps`-File is available):

```
\begin{figure}[htbp]
  \begin{center}
    \includegraphics[width=0.7\textwidth]{picture.eps}
  \end{center}
  \caption{This is a screenshot of my \LaTeX{} environment.}
  \label{pic:my_first_picture}
\end{figure}
```

Comments:

- The *placement* parameters (`[htb]`) specify that the figure may appear **h**ere, at the **t**op or **b**ottom of the current page, or at a separate **p**age by itself. If you omit any of these placement permissions, the figure will not be placed in that way. This is a matter of writing style—for most purposes, `[htb]` should be fine. If figures are slightly too large (usually wider than the page), they are placed at the end of the chapter or document, on separate pages. Since the order of figures is never changed, all following figures appear after the initial large one. When this happens, you should decrease the figure in size (i.e., change the `width`-parameter). You can also try to add a `!` to the placement parameter (e.g., `[!htb]`). This relaxes some constraints that LaTeX puts on the placement of figures and makes things easier sometimes.
- The width of the picture can be specified *absolutely* (`5cm, 2.8in`, etc.) or *relatively* as in the example. You can also specify the *height* using `height`, but only one of the two parameters should be used unless you want to change the aspect ratio.
- You can give a caption to the figure which will appear below the figure. You can format the caption just like any LaTeX-text (e.g., by including `\emph{}`-commands).
- The label is useful for referring to the picture number with, e.g., `~\ref{pic:my_first_picture}` or the page on which the picture was placed with, e.g., `~\pageref{pic:my_first_picture}`.
- If you add the `draft` parameter to the package (`\usepackage[draft,dvips]{graphicx}`), only the bounding boxes of figures will be displayed, which saves a lot of time when viewing documents with `xdvi`.

The above example generates the following output (Figure 1, Page 6)

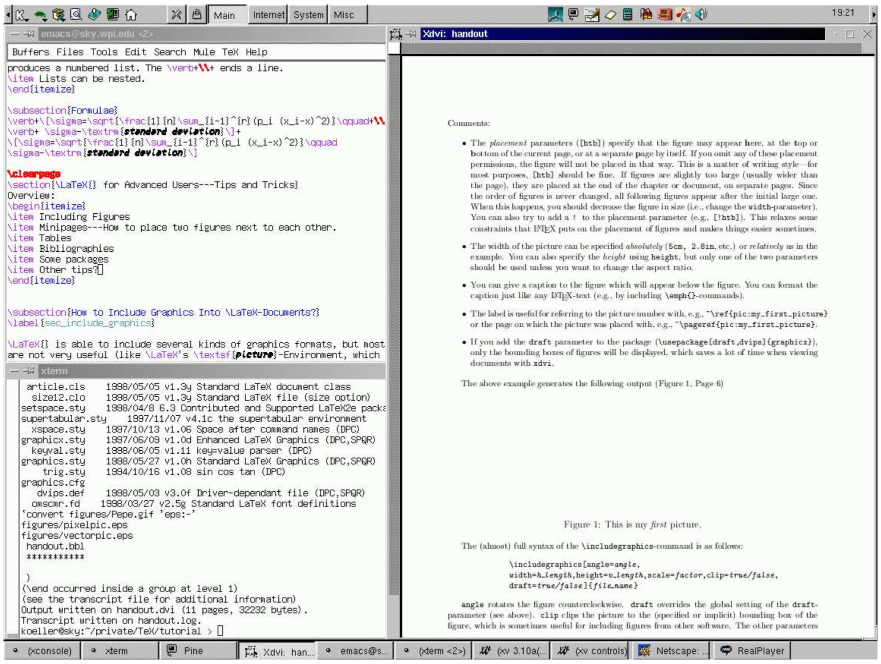

Figure 1: This is a screenshot of my LaTeX environment.

The (almost) full syntax of the `\includegraphics`-command is as follows:

```
\includegraphics[angle=angle,
width=h_length,height=v_length,scale=factor,clip=true/false,
draft=true/false]{file_name}
```

`angle` rotates the figure counterclockwise. `draft` overrides the global setting of the `draft`-parameter (see above). `clip` clips the picture to the (specified or implicit) bounding box of the figure, which is sometimes useful for including figures from other software. The other parameters should be self-explanatory. If you don't need any of the parameters, you can use the simple `\includegraphics{filename.eps}` syntax.

The file `epslatex.ps`, provided on the CS-GSO homepage, goes into great detail about the usage of graphics in LaTeX. It handles dozens of complicated cases of the inclusion of figures, such as the customization of captions, the placing of tables next to figures, etc. Please refer to that documentation if you have greater plans with figures, but please view it on the screen (`ghostview`) and don't print it, since it has 86 pages.

4.2 Minipages—Placing Two Figures Next To Each Other

If you have little space in your document, you may want to place two figures next to each other in the text. This is accomplished by enclosing each `\includegraphics`-command into a "minipage". You have to set the widths of the minipages to values that together do not exceed the page width (i.e., if you use `\textwidth`, the factors have to add up to less than 1). You can scale the figures inside each minipage relatively to the minipage (i.e., the `\textwidth`-function *inside* a minipage-environment returns the width of the minipage, not the whole page). If you need extra space between the two figures, decrease their combined width to something not larger than 90% of the text width and include a `\hfill` between the minipages. The two Figures 2 and 3 on page 8 were included with the following source code [1]:

```
\begin{figure}[htb]
  \begin{minipage}{0.45\textwidth}
    \begin{center}
      \includegraphics[width=0.9\textwidth,angle=270]{pixelpic.eps}
      \caption{A Bitmap Picture.}
      \label{pic_bitmap}
    \end{center}
  \end{minipage}
\hfill
  \begin{minipage}{0.45\textwidth}
    \begin{center}
      \includegraphics[width=0.9\textwidth,angle=270]{vectorpic.eps}
      \caption{The Same Picture as a Vector Graphic.}
      \label{pic_vector}
    \end{center}
  \end{minipage}
\end{figure}
```

[1] Here, the width of the two minipages together is $(0.45+0.45)*$`\textwidth`; the space between the figures inserted by `\hfill` therefore amounts to 10% of the width of the page.

4.3 Bitmaps vs. Vector Graphics

A vector graphic picture can be scaled without loss of quality and its file is much smaller. So unless you *need* a bitmap (screenshots), you should always convert your original picture into a vector graphic. The difference between the same picture inserted as a bitmap and a vector graphic, respectively, is shown in Figures 2 and 3. Also, the `eps`-file of the bitmap picture is 35KB large, vs. 13KB for the vector graphic file. As an *uncompressed* PostScript-File, the size of the bitmap picture exceeds 4MB!

Figure 2: A Bitmap Picture.

Figure 3: The Same Picture as a Vector Graphic.

The best way of creating EPS-Figures is exporting them with a software that supports EPS, such as *Corel Draw* in any version. You can also convert figures from any Windows software (e.g., Powerpoint) through Corel Draw. If no such software is available, there are a few ways to convert, described in the LaTeX help on the CS-GSO homepage. Drawing tools on UNIX include `xfig` and `gnuplot`, the former for general drawing, the latter for function plotting.

4.4 Tables in LaTeX

Tables are created with the `tabular` environment. When you create this environment you specify how many columns to have and how the contents are to be aligned (use `l`, `c` or `r` to represent each column with either left, center or right alignment) and where you want vertical lines (use `|`). The contents of the columns are separated by a `&` and rows by `\\`. Here's a simple example

```
\begin{tabular}{l|c|r}
left & centre & right\\
more left & more centre & more right\\
\end{tabular}
```

left	centre	right
more left	more centre	more right

To draw a full horizontal line, use `\hline` otherwise draw a line across selected columns using `\cline`. The `\multicolumn` command allows items to span columns. It takes as its first argument the number of columns to span. The following, more complicated example shows how to use these facilities.

```
\begin{tabular}{||l|lr||} \hline
\textbf{Veg}  & \multicolumn{2}{|c||}{\textbf{Detail}}\\\hline
carrots    & per pound & \pounds 0.75 \\ \cline{2-3}
           & each      &         20p  \\ \hline
mushrooms  & dozen     &         86p  \\ \cline{1-1} \cline{3-3}
toadstools & pick your own & free     \\ \hline
\end{tabular}
```

Veg	**Detail**	
carrots	per pound	£0.75
	each	20p
mushrooms	dozen	86p
toadstools	pick your own	free

If the text in a column is too wide for the page, LaTeX won't automatically text-wrap. Using `p{5cm}` instead of `c`, `l` or `r` in the tabular line will wrap-around the text in a 5 cm wide column.

Tables won't continue on the next page if they're too long. The `supertabular` commands are needed to do this. See the document `supertab.ps` from the CS-GSO homepage for details and examples[2].

[2]From the `http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/` website.

4.5 Bibliographies

You can maintain a separate file which contains all your references and use the `bibtex` program to create bibliographies automatically. Some literature databases on the Web provide `bibtex`-formatted entries for each paper.

A few typical `bibtex`-Entries (in a file called `mybibliography.bib`):

```
@inproceedings{gupta,
  author =  {A. Gupta and V. Harinarayan and D. Quass},
  title =  {Aggregate-Query Processing in Data Warehousing
                Environments},
  year =  {1995},
  booktitle =  {Proceedings of VLDB},
  month =  {June}
}

@article{colby,
  author =  {L.S. Colby and A. Kawaguchi and D.F. Lieuwen and
                I.S. Mumick and K.A. Ross},
  title =  {Supporting Multiple View Maintenance Policies},
  year =  {1996},
  journal =  {{AT\&T} Technical Memo}
}
```

You can use these entries as follows:

`Gupta~\cite{gupta} and Colby~\cite{colby} wrote papers on the matter.`

which gives the following output:

Gupta [GHQ95] and Colby [CKL$^+$96] wrote papers on the matter.

You include the bibliography in your paper as follows:

```
\bibliographystyle{alpha}
\bibliography{mybibliography}
```

References

[CKL$^+$96] L.S. Colby, A. Kawaguchi, D.F. Lieuwen, I.S. Mumick, and K.A. Ross. Supporting multiple view maintenance policies. *AT&T Technical Memo*, 1996.

[GHQ95] A. Gupta, V. Harinarayan, and D. Quass. Aggregate-query processing in data warehousing environments. In *Proceedings of VLDB*, June 1995.

To update the bibliography, run `bibtex mydocument` from the command line, with `mydocument` the file name of your main LaTeX source file. Valid bibliography styles are `alpha,plain,abbrv,unsrt`. You can define your own styles. IEEE, ACM, and SIAM have their own bibliography styles (`ieeetr,acm,siam`). All formatting necessary for their bibliography requirements is done automatically in LaTeX/BibTeX!

4.6 Classes and Packages

4.6.1 Classes

LaTeX base classes include `article, report, letter` and do what they say. Other classes (`*.cls`-files) are available from many publishers, so adhering to a typesetting standard given by a publisher (most journals) is very easy in LaTeX. Templates for the basic three classes are given on the CS-GSO homepage.

4.6.2 Packages

Packages are included with the `\usepackage{}` command directly after the `\documentclass{}` command. If you want to include more than one package, separate the names by commas or use multiple `\usepackage`-commands. Some of the more important packages are:

`setspace` allows to freely adjust spacing between lines without affecting footnotes. Doublespacing is invoked with `\doublespacing` or `\setstretch{1.6}` in the preamble of the document.

`graphicx` allows to include PostScript and Encapsulated PostScript figures into LaTeX documents.

`xspace` a device to fix the problem of command names swallowing up the blank that follows them (try typing "`\LaTeX is cool.`" on a standard LaTeX system). If you define your own text-printing macros using the `\newcommand`-command, you should end each command definition with an `\xspace`-command.

`supertabular` is a package that allows to create tables that extend over several pages. See the CS-GSO homepage for details.

`amsfonts` provides many fonts that are useful in more complicated mathematical environments, such as the famous "blackboard bold" letters ($\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$)

`eufrak` provides old German *fraktur* letters ($\mathfrak{Andreas\ Koeller}$, but for mathematical purposes)

`amssymb` provides a few hundred extra symbols, the LaTeX equivalent of a "WinDings" font (e.g., $\pitchfork, \lessdot, \ltimes, \multimap, \ldots$).

`ifthen` provides Boolean operators and loop commands to do some rudimentary programming (e.g., useful for automatically generating lists).

Many standard packages are documented on `http://www.math.uncc.edu/tex/texstuff.html`, others are available on the Web. Probably every conceivable problem with typesetting has been solved in LaTeX with such a package. The American Mathematical Society (`http://www.ams.org/tex`) provides a number of packages that are very useful for more math-heavy printing.

4.7 Customization and Fine-tuning—LaTeX-macros

You can define your own macros (commands) in LaTeX and even your own environments (e.g., your own way of printing lists). Even proramming language constructs like Boolean operators, if/then-statement and while/for-loops are available (with the `ifthen`-package). Some examples:

Definition	Usage	Prints
A command with no parameters:		
`\newcommand{\xvec}{x_1,\ldots,x_n}`	`$\xvec$`	$x_1, \ldots, x_n$
Simple abbreviations of long commands:		
`\newcommand{\lra}` `{\ensuremath\Longleftrightarrow}`	`$A\lra B$`	$A \Longleftrightarrow B$
A command with parameters:		
`\newcommand{\subxvec}[3]` `{\ensuremath{#1_#2.\ldots,#1_#3}}`	`\subxvec{a}{i}{j}`	$a_i. \ldots, a_j$
A new environment for "quotes in small print and smallcaps font":		
`\newenvironment{smquote}` `{\begin{quote}\small\scshape}` `{\end{quote}}`	`\begin{smquote}` `This is\\` `A Quote.` `\end{smquote}`	THIS IS A QUOTE.
Special characters:		
`\newcommand{\bs}{\texttt\symbol{92}}`	`A backslash is a \bs.`	A backslash is a \\.
A "tab" for writing source code in non-`verbatim`-mode:		
`\newcommand{\T}{\hspace*{5mm}}`	`first line` `\T second line` `third line`	first line second line third line

4.8 Additional Fonts and Encodings

LaTeX provides more than the 5 standard fonts through the NFSS (New Font Selection Scheme). Also, special letters for practically all Latin-written languages are included and available through the *T1 encoding* (ăąćčďěęğĺľłńňŋőŕřśšşťţűůÿźžżijàáâãäåæçèéêëìíîïðñòóôõöœøùúûüýþß). Other languages can be typed in LaTeX as well. Cyrillic fonts are built-in to modern distributions. Other writing systems such as Arabic and Devanagari, as well as a few other Indian scripts are available but have to be installed separately.

Examples:

Сегодня, 7 февраля 2000

Thiſ iſ ancient Gothic font.

This font is called Dunhill.