LaTeX: from beginner to TeXpert

John Gardner

This version: September 2008

Contents

1 Introduction

This document introduces the LaTeX typesetting system. After digesting the information below, you'll be able to:

- Download and install LaTeX on your PC or Mac
- Create basic documents using LaTeX
- Install new LaTeX packages
- Insert tables and figures into a LaTeX document
- Use LaTeX's cross-referencing, footnote and basic bibliography features
- Insert equations into a LaTeX document

These topics cover the majority of tasks that most people need to do when writing a document. However, please note that while the LaTeX system makes it very easy to create professional-looking documents, it is both comprehensive and extensible. There are many topics that are not covered by this basic tutorial.

Fortunately, LaTeX is very well documented. If you come across something that you can't figure out how to do, ask your old friend Google for help.

2 What is LaTeX?

At its core, LaTeX is a typesetting system that allows authors to create highly polished documents without having to worry about formatting, page breaks, object positioning, or any other style concerns that distract them from focusing on writing. LaTeX is pronounced "lay-tech," as it is an extension of TeX ("tech"), the original typesetting system. You can read all about the history of TeX and LaTeX on Wikipedia.

LaTeX is used widely in a variety of professions. Mathematicians, physicists, economists, statisticians and other academics and professionals that regularly use mathematical notation in their documents often use LaTeX because of the ease with which it handles such notation. Many publishers use TeX-based systems for typesetting documents.

3 How does LaTeX work?

LaTeX differs from traditional word processors in two fundamental ways:

1. Generally, LaTeX documents are written using the easy-to-learn LaTeX markup language, rather than by using a graphical interface to apply styles [1]

2. LaTeX processes your document after you have entered your text. So unlike word processors, it can use information about the total length of your document, number of tables, etc. to find the optimal places for tables, figures, page breaks, etc. to format your text

The following is an example of a very basic LaTeX document:

```
\documentclass{article}
\author{Your Name}
\title{Test Document}
\begin{document}
```

[1] Graphical editors, such as Scientific Word (a commercial application) and LyX (an open-source application), are available; these applications are easier to use if you know how LaTeX works, so it's a good idea to learn it even if you don't plan to write LaTeX markup by hand.

```
\maketitle
This is a test document
\end{document}
```

With any LaTeX distribution, saving the above text as a `.tex` file and running LaTeX on that file would produce the following:

Test Document

Your Name

April 21, 2006

This is a test document.

LaTeX is designed to create the same output on any system. As a result, if you distributed the above text to anyone with a working LaTeX distribution, regardless of their particular system, they would get the exact same result. LaTeX outputs compiled documents in several formats, but the most popular is PDF.

4 Getting LaTeX

All you technically need to create LaTeX documents is a LaTeX engine – the binary files and libraries that will convert plain text `.tex` files to polished PDF files. LaTeX can be run from the command line, so Linux and DOS aficionados will feel right at home. However, using a frontend for LaTeX can make things much easier. Most frontends are essentially text editors with functions to

- Compile documents with LaTeX without using the command line
- Facilitate writing in the LaTeX language (wizards for table creation, syntax highlighting, code completion, etc.)

In this document, I assume that you'll use both a LaTeX engine and a frontend. There are many engines and frontends to choose from on every operating system. I'm going to describe how to install the most popular (and easy to

install) open-source tools for Windows and Mac OS X. Other LaTeX tools have different configuration requirements and operating instructions, but almost every working environment involves (i) editing raw `tex` files using a frontend and (ii) compiling the LaTeX document to a PDF, generally using buttons or menu commands in the frontend rather than the command line.

4.1 On Windows

Engine MikTeX is a popular open-source distribution. To install, visit `http://www.miktex.org`, download the executable, and follow the dialog. Additional installation instructions are on the download page.

Frontend TeXnic Center, available from `http://toolscenter.org`, is an open-source frontend with many helpful features. Installation is standard, just download and open the executable, which opens a wizard.

TeXnic center is automatically configured to work with MikTeX. To test out your setup, save the sample document above as a `tex` file using TeXnic Center and select `Build` ⇒ `Current file`. If everything is set up properly, a new PDF file (along with a log file) will be created in the directory where your document is saved.

4.2 On Mac OS X

Engine gwTeX is a free and open-source LaTeX distribution for OS X that comes with a graphical installer. To install, download the i-Installer application, select a mirror, then select the TeX package. Additional installation instructions are available at `http://ii2.sourceforge.net/tex-index.html`. Once installation is complete, all you need is a frontend.

Frontend TeXShop (`http://www.uoregon.edu/~koch/texshop/`) is a very popular LaTeX frontend for OS X. Installation requires a simple drag and drop to the ~/Applications folder. TeXShop is automatically configured to work with gwTeX, so if that's the engine that you're using, you're set.

To test out your distribution, try saving the sample document above as a `tex` file and running LaTeX on your document by pressing command-t. If everything is configured properly, a window will appear similar to the example

output above, and a new PDF file (as well as a log file) will appear in the directory where your file is saved.

4.3 On Linux

Different Linux systems have their own application management utilities (apt-get or rpm, for example), and installation will depend on your particular Linux distribution. Ubuntu users can use the Synaptic Package Manager. Kile is a popular and easy-to-use frontend that works with both KDE and Gnome.

5 LaTeX basics

5.1 LaTeX commands

LaTeX commands generally begin with a backslash and take the form:

```
\command[options]{argument}
```

For example,

```
\section{Introduction}
```

would define a new section, named "Introduction." The "%" character defines a comment, and everything from that character to the end of the line is commented out and will be ignored by LaTeX. To insert the "%" character into a document, escape it with a backslash: `\%`. To insert a backslash, use `$\backslash$`. This is true of other characters as well; use `\&` to produce &, `\$` to produce $, `\#` to produce #, `\_` to produce _ and `\{` to produce {.

Quotes work a bit differently in LaTeX. To insert quote marks, use the form ``` ``text'' ```. That is, the ` character (top left of the keyboard) twice, followed by the single quote character, ', twice.

5.2 The preamble

Everything before the line `\begin{document}` is part of the preamble. A typical preamble might look like this:

```
\documentclass{article}
\usepackage{graphicx}
\usepackage{amsmath,amsthm,amsfonts}
\title{Test}
\author{Test}
\date{}
```

In the example above:

- `\documentclass{article}` tells LaTeX that the document is an article. Other classes include book, letter and slides
- `\usepackage{graphicx}` tells LaTeX to use the graphicx package, which allows users to include many types of graphics in their documents. Packages are covered later on. The `\usepackage{amsmath,...}` command invokes packages from the American Mathematical Society that extend the functionality of LaTeX
- `\title{}` and `\author{}` obviously define the title and author
- `\date{}` tells LaTeX to leave the data blank. `\date{April 2008}` would print "April 2008" as the date. Leaving the `\date{}` line out would cause LaTeX to use today's date

The `\documentclass{}` command has options. For example,

```
\documentclass[11pt,twocolumn]{article}
```

would organize body of the document into two columns. Note that options are separated by a comma. Other options include:

- `oneside` or `twoside`: change the margins for a one or two-sided document
- `landscape`: change the document from portrait to landscape
- `titlepage` or `notitlepage`: define whether there is a separate title page, or if the title, author and date are presented at the top of the article

There are document classes other than article. The book class, for example, is useful for writing books. By installing the beamer package, you can use the beamer document class to make impressive presentations in LaTeX. For information on other document classes, consult Google.

5.3 The document body

Everything after the preamble and between `\begin{document}` and `\end{document}` is part of the document body. Most of a LaTeX document is simply plain text. To start a new paragraph, insert two carriage returns (blank lines). LaTeX will ignore one blank line. To force a line break, use `\\`.

5.4 Document structure

A document's structure is defined using `\section{}` commands. LaTeX is strongly based on well-structured documents. The structure tags include:

- `\section{Name}`
- `\subsection{Name}`
- `\subsubsection{Name}`
- `\paragraph{Name}`

To insert an unnumbered section, use the command `\section*{Name}`. The section numbering will continue as normal with the next section, subsection, etc.

The `\paragraph{}` command doesn't need to be included unless you want to insert a heading for a paragraph. The image below shows the different structure commands in use:

1 Section command

Section star command

This section is not numbered.

2 Section command

Text here. The numbering continues normally.

2.1 Subsection command

Text here.

2.1.1 Subsubsection

Paragraph command This paragraph has a title.

6 Environments

Environments are special blocks of text. For example, the itemize and enumerate environments create bulleted and numbered lists, respectively. The following markup:

```
\begin{itemize}
    \item First thing
    \item Second thing
    \item Third thing
\end{itemize}

\begin{enumerate}
    \item First numbered thing
    \item Second numbered thing
\end{enumerate}
```

would produce a bulleted list followed by a numbered list.

Note that environments always begin with `\begin{environmentname}` and end with `\end{environmentname}`. They can be nested, so one item of a bulleted list might contain another bulleted list, or a numbered list, etc.

Other frequently used environments include:

Quote `\begin{quote}...\end{quote}` creates a section of indented, quoted text

Verbatim `\begin{verbatim} ... \end{verbatim}` is similar to `<pre>` in HTML. In the verbatim environment, text is printed in a monospace font and special characters (such as \ and %) are ignored. Verbatim is useful for typing code tips

Description Description lists are similar to bulleted lists, with a bold item name followed by a description:

```
\begin{description}
    \item[First item] Description of item
    \item[Second item] Description of item
\end{description}
```

This list is presented using the description environment; the code samples in this document are presented using the verbatim environment.

7 Modifying text styles

A nice feature of LaTeX is that it absolves the author of most formatting duties. Nevertheless, it's still occasionally necessary to manually enter certain text formatting.

- To insert bold text, use `\textbf{text here}`
- To insert italic text, use `\emph{text here}`
- To insert monospace text, use `\texttt{text here}`
- To use verbatim text within a sentence, use `\verb|your text here|`. Note that any delimiter can be used, for example `\verb+your text here+` will produce the same result
- To center a line, you can use

```
\begin{center}
Your text here
\end{center}
```

8 Packages

Packages extend LaTeX's functionality. Package installation essentially consists of two steps (after downloading the package, that is):

1. Running LaTeX on the `ins` file to produce `sty` or `cls` files
2. Copying the newly created files to an appropriate directory and updating the LaTeX database

However, there are exceptions. The filetypes `sty` and `cls` stand for style and class, respectively. If a package does not come as an `ins` file, but rather a `sty` or `cls` file, it does not need to be processed with LaTeX, and you can skip directly to step two. Also, running LaTeX on an `ins` file usually produces a `dtx` file. This file can be processed with LaTeX to create a manual for the package.

Note To process a package file (`ins` or `dtx`) with LaTeX, just open that file with your frontend and process it like you would a normal `tex` file.

Hopefully, you'll never need to do any of this. Most LaTeX distributions include automatic, graphical methods for finding and installing packages (including gwTeX and MikTeX).

Windows The easiest way to install a package on a PC using MikTeX is to use the MikTeX package manager, which is available through the Start Menu. Just open the package manager, select a mirror (the location from which you will download files), and navigate to the package that you want to install. MikTeX will take care of the rest. Another nice feature of MikTeX is that if you are processing a `tex` file that requires a package that isn't installed on your machine, it will prompt you to download it.

OS X Launch i-Installer to browse through the available packages. Almost every popular package is installed by default with gwTeX, so you may never need to do this. To install a package on your Mac that isn't available using i-Installer, process the files as described above, and move the `cls`, `sty` and other files to ~/Library/texmf. If this directory does not exist, create it.

Next, I discuss some popular packages. These packages are already installed with gwTeX and MikTeX, so there is no need to download and install them.

8.1 The graphicx package

The graphicx package allows you to insert images into a LaTeX document. To use it, first use the command `\usepackage{graphicx}` in your document preamble. Then, to insert a graphic, use the command:

```
\includegraphics[options]{filename.png}
```

The graphicx package supports many filetypes, including PDF, PNG and JPG. The options include:

- width=Xin
- height=Xin
- scale=X (where x is between 0 and 1)

You can also use, for example, the command `width=.8\textwidth` to scale a picture to 80% of the width of your text. If you only use, say, the `width` option, the height will be scaled proportionally; if you want to skew your image, you need to specify both the height and width.

8.2 The geometry package

While formatting documents using LaTeX is easy, changing those default formats can be fairly difficult. The geometry package can make changing certain aspects of your document, including the margins, much easier. To change the margins to 1" all around, for example, use:

```
\usepackage[margin=1in]{geometry}
```

in your document's preamble.

8.3 The AMS packages

The American Mathematical Society provides packages to make typesetting mathematical documents easier. To use these packages, include the command

```
\usepackage{amsthm,amsmath,amsfonts}
```

in the preamble of your document. The `amsthm` package allows you to include Theorems, Propositions, Proofs, etc. in your document. The `amsfonts` package includes symbols that are not available in standard LaTeX—such as $\mathbb{R}$ (blackboard bold)—which I inserted using the command `$\mathbb{R}$`. The `amsmath` package includes several environments and commands for displaying mathematics. These commands, which are introduced in a later section, make certain mathematical constructs much easier to typeset.[2]

8.4 Other packages

For just about every modification that you might want to make to a standard LaTeX document, there is a pre-made package to help you do so. For example, in this document, I have used the `fourier` package to change the font, the `hyperref` package to make links clickable, and the `url` package to make URLs pretty. To learn more about the packages described, or to download new packages, visit the Comprehensive TeX Archive Network (CTAN).[3]

9 Figures and tables

Figures and tables are LaTeX environments with special attributes, such as the `\caption{}` command, which gives them titles within the document. They are called float elements because their position in the final compiled document depends on LaTeX's style algorithm.

9.1 Figures

To insert a figure, use

```
\begin{figure}[hbtp]
\caption{Figure name}
\begin{center}
\includegraphics{filename.pdf}
\end{center}
\label{your-reference-key}
\end{figure}
```

[2] See the AMS-LaTeX page (`http://ams.org/tex/amslatex.html`) for complete documentation.

[3] `http://ctan.org`.

In the above markup,

- `\begin{figure}` simply tells LaTeX that there is a figure environment
- `[hbtp]` determines how LaTeX will place the figure (here (h), bottom (b), top(t), page(p)). LaTeX will first attempt to insert the figure at its insertion point in the `tex` file. If this is not possible due to space or other aesthetic considerations, it will try to place it at the bottom of the page, then at the top of the page, then on a special page reserved just for float elements. The order in which h, b, t and p are specified determines where LaTeX tries to place the float first. To force the graphic to appear in its original place, for example, you could put `\begin{figure}[h]`, omitting b, p and t
- `\caption{Figure name}` specifies the name of the figure
- `\begin{center}` simply tells LaTeX to center the figure on the page. Don't forget to end the centering environment before you end the figure environment
- `\includegraphics{...}` specifies the location of the file that is being inserted as a figure
- `\label{your-reference-key}` is a label that you can use to refer to the figure in the text. For example, if you label your figure "fig1" then you can reference it later on by typing `\ref{fig1}`

9.2 Tables

A floated table in LaTeX consists of two environments: "table," the actual floated entity in the text, and "tabular," the data contained in the table. For example,

```
\begin{table}[hbtp]
\caption{This table is an example}
\begin{center}
\begin{tabular}{c|cc}
First row, first column &
  First row second column &
  First row, third column \\ \hline
Second row, first column &
  Second row, second column &
```

```
  Second row, third column \\
Third row, first column &
  Third row, second column &
  Third row, third column \\
\multicolumn{3}{c}{...}
\end{tabular}
\end{center}
\label{exampletable}
\end{table}
```

would produce Table 1.

Table 1: This table is an example

First row, first column	First row second column	First row, third column
Second row, first column	Second row, second column	Second row, third column
Third row, first column	Third row, second column	Third row, third column
	...	

Everything except the code between `\begin{tabular} ... \end{tabular}` is the same as the figure environment described above. Here's how the tabular environment works:

- `\begin{tabular}{c|cc}` tells LaTeX to start a new tabular environment with three centered columns. The bar ("`|`") after the first "c", tells LaTeX that the first column has a vertical border. Using `{lcrr}` would create four columns, the first left aligned, the second centered, and the third and fourth right aligned
- Table cells are separated by "&" and table rows are separated by "`\\`"
- `\hline` creates a horizontal line
- `\multicolumn{3}{c}{Text here}` creates a row that spans all three columns, is centered, and contains the text "Text here"

There are more complicated options for creating and inserting tables, but the rules above cover the commands needed to create most basic to intermediate tables.[4]

10 Annotations

LaTeX is capable of automatically creating important annotations, such as footnotes, cross references, tables of contents and bibliographies. Note that, since the following commands require LaTeX to automatically number text elements, LaTeX must be run on your document twice for proper display.

10.1 Footnotes and endnotes

To insert a footnote, simply type `\footnote{text here}`. LaTeX will automatically insert the footnote number and text.[5]

To use endnotes, first invoke the endnotes package in the preamble:

```
\usepackage{endnote}
```

and use the command `\endnote{Text}` to create new endnotes. At the point in your document where you want the endnotes to appear, simply type `\theendnotes`.

10.2 Cross references

To reference a labeled Equation, Table or Figure, use `\ref{your-reference-key}` where "your-reference-key" is the argument to the `\label{your-reference-key}` command in the table or figure environments.

10.3 Table of contents

To insert a table of contents, simply put `\tableofcontents` at the beginning of your document.

[4] MS Office users can try Excel2LaTeX (`http://www.ctan.org/tex-archive/support/excel2latex/`), which does the same thing using Excel spreadsheets. OpenOffice users can use Calc2LaTeX (`http://calc2latex.sourceforge.net`) to convert between Calc spreadsheets and LaTeX tables. Both utilities are cross-platform.

[5] To create an "attribution" footnote, where the first footnote is marked by an asterisk, use the \thanks{text here} command.

10.4 Bibliography

To create a bibliography, insert a list of the citations at the end of your document, using the form:

```
\begin{thebibliography}{99}
...
\bibitem{key1} Gardner, John. 2006.
``\LaTeX{}: from beginner to \TeX pert.''
\emph{Dataninja}. Available online at
\texttt{http://generaldisarray.wordpress.com}.
...
\end{thebibliography}
```

You must manually type the bibliography entries. To refer to an item within the text, use `\cite{key}`[1]. The `{99}` tells LaTeX that there a maximum of 99 entries in the bibliography. LaTeX needs to know this so it can correctly justify the bibliography entries with their numbering on the left.

A more efficient way to create bibliographies is to use BibTeX, which allows you to maintain a database of citations and call them as needed in your bibliography. There are also graphical tools for managing your reference databases, so you don't have to hard code the citations, and can easily change them to different formats. However, BibTeX is too complicated to explain in this document. For an introduction, see `http://bibtex.org`.[6]

11 Inserting mathematics

There are several ways to include mathematical notation in LaTeX documents.

11.1 Inline math

To include some mathematical notation within a paragraph, without offsetting from the rest of the text, enclose the notation between dollar signs. For example, `$a^2+b^2=c^2$` would produce $a^2 + b^2 = c^2$.

[6] To insert references that look like Author (2008) or (Author 2008), use the natbib package.

11.2 Display math

The displaymath environment lets you offset some mathematical notation from the rest of the document. The code

```
\[
    a^2+b^2=c^2
\]
```

would create a paragraph break and center the equation on the page, like this:

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

11.3 Equation

The equation environment can be used to place numbered equations in the text. For example,

```
\begin{equation}\label{pythag}
    a^2+b^2=c^2
\end{equation}
```

would offset the equation just like the displaymath version did, but it would have a number in parenthesis on the right, and you would by able to call it in the text by typing, for example, "as we see in equation `\ref{pythag}...`"

11.4 Align

The align environment allows you to align parts of equations at the equal sign. For example,

```
\begin{align} \label{myeqn}
    a &=b+c \\
    d &=e+f
\end{align}
```

would produce

$$a = b + c \tag{1}$$
$$d = e + f \tag{2}$$

Notes You must have invoke the `amsmath` package in order to use the align environment. Also, it is possible to suppress the numbering by using the commands `\begin{align*}` and `\end{align*}`. Labeling works the same for the align and equation environments. A shortcut to referencing an equation is the command `\eqref{label}`. For example, `\eqref{myeqn}` produces (1).

11.5 Mathematical notation

There are many commands for inserting specific mathematical operators and symbols into equations. They can all be found online, and as always, use Google if you can't figure out a specific command. The following are some common operators and commands.

Greek letters Generally, just use the spelled-out letter. For example, `\beta` produces β and `\epsilon` produces ϵ. For upper case, capitalize the letter: `\Gamma` produces Γ while `\gamma` produces γ.

Misc. symbols LaTeX can produce almost any symbol; the following are just a few examples. `\leftarrow` produces $\leftarrow$. Use `\Leftarrow` for a double arrow: $\Leftarrow$. Similarly, `\rightarrow` produces $\rightarrow$ and `\Leftrightarrow` produces $\Leftrightarrow$. For $\leq$, $\geq$, use `\leq` (less than or equal to), `\geq` (greater than or equal to).

Indexing and exponents Subscripts are denoted using the underscore (`x_i`) and superscripts use the `^` key (`a^2`). To type "$i_{j,k}$" you need to write `i_{j,k}` to tell LaTeX that the "j,k" comprises the entire subscript. The bracket characters are generic grouping operators in LaTeX, and they won't appear in your document.

Some operators `\sum{1/x}` produces $\sum 1/x$ and `\sum_{i=1}^{\infty}{x_i}` produces $\sum_{i=1}^{\infty} x_i$. Other operators include `\prod` ($\prod$), `\int` ($\int$), `\sin`, `\log`, `\arg`, `\max` (combine them as `\arg\max_x f(x)` to get $\arg\max_x f(x)$), `\lim`, `\sqrt`, etc. To produce, for example, a cube root, use the command `\sqrt[3]{x}` ($\sqrt[3]{x}$). Note that, for operators, LaTeX automatically chooses the appropriate location for arguments and subscripts. For example, if `\int_0^1 f(x) dx` is inline, it looks like $\int_0^1 f(x)dx$, whereas if it is displayed math, it looks like

$$\int_0^1 f(x)dx.$$

Decorations `\hat{x}` puts a hat over x ($\hat{x}$), as does `\hat x` (use the former if there are multiple characters under the hat). Similarly, `\tilde{x}` produces $\tilde{x}$, `\bar{x}` produces $\bar{x}$, `\underline{x}` produces $\underline{x}$, `\vec x` produces $\vec{x}$, etc. You can use the `\overset` and `\underset` commands to place labels over symbols. For example, `x\overset{p}{\rightarrow}b` produces $x \overset{p}{\rightarrow} b$.

Fractions Use `\frac{a}{b}` to display $(\frac{a}{b})$.

Brackets For brackets use "(", "[" or `\{` and `\}` for "{" and "}". However, if you need stretched brackets, use `\left(...\right)` or `\left\{ <math here> \right\}`. For example,

```
\left[ \frac{1}{2} \right]
```

produces

$$\left[\frac{1}{2}\right].$$

Matrices To insert a matrix with square brackets, use

```
\[
\begin{bmatrix}
    a       & b         & \cdots        & c \\
    \vdots  & \vdots    & \ddots        & \vdots \\
    d       & e         & \cdots        & f
    \end{bmatrix}.
\]
```

Once again, you must invoke the `amsmath` package The code shown above would produce:

$$\begin{bmatrix} a & b & \cdots & c \\ \vdots & \vdots & \ddots & \vdots \\ d & e & \cdots & f \end{bmatrix}.$$

Other matrix types include `pmatrix` for a matrix with parenthetical brackets and `cases` to define, for example, a piecewise function. Notice, in the above, the various "dots" commands. Another useful one is `\ldots` which inserts an elipsis ... The reason for the command, rather than simply typing three periods, is that LaTeX precisely controls the spacing of the elipsis according typographical standards.

Other tips To insert a space within an equation, use `\quad` for a space the width of an "M" or `\qquad` for twice that much space. To put some ordinary text within a math environment, use `\text{}`. For example:

```
y_i=\alpha+\beta x_i+\varepsilon_i \qquad e_i \sim N(0,\sigma^2)
\quad \text{A basic line}
```

would produce:

$$y_i = \alpha + \beta x_i + \varepsilon_i \qquad e_i \sim N(0, \sigma^2) \quad \text{A basic line}$$

For help with other symbols and operators, see the American Mathematical Society's Short Math Guide for LaTeX.[7]

12 For further reference

The instructions above cover many of the basic functions of LaTeX, but there are many more. A good, thorough introduction is The LaTeX Primer (PDF). I have other tools and references on my data analysis weblog Dataninja.

References

[1] Gardner, John. 2006. "LaTeX: from beginner to TeXpert." *Dataninja*. Available online at `http://dataninja.wordpress.com`.

[2] Indian TeX Users Group. 2003. "LaTeX Tutorials: A primer." Available online at `http://sarovar.org/frs/download.php/120/ltxprimer-1.0.pdf`.

[7] `ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf`.