The Not So Short Introduction to $\LaTeX 2_\varepsilon$

Or $\LaTeX 2_\varepsilon$ in 95 minutes

by Tobias Oetiker

Hubert Partl, Irene Hyna and Elisabeth Schlegl

Version 3.20, 09 August, 2001

Thank you!

Much of the material used in this introduction comes from an Austrian introduction to LaTeX 2.09 written in German by:

Hubert Partl `<partl@mail.boku.ac.at>`
Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna `<Irene.Hyna@bmwf.ac.at>`
Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl `<no email>`
in Graz

If you are interested in the German document, you can find a version updated for LaTeX 2_{ε} by Jörg Knappen at
`CTAN:/tex-archive/info/lshort/german`

While preparing this document, I asked for reviewers on `comp.text.tex`. I got a lot of response. The following individuals helped with corrections, suggestions and material to improve this paper. They put in a big effort to help me get this document into its present shape. I would like to sincerely thank all of them. Naturally, all the mistakes you'll find in this book are mine. If you ever find a word which is spelled correctly, it must have been one of the people below dropping me a line.

Rosemary Bailey, Friedemann Brauer, Jan Busa, Markus Brühwiler, David Carlisle, José Carlos Santos, Mike Chapman, Christopher Chin, Carl Cerecke, Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes, David Dureisseix, Elliot, David Frey, Robin Fairbairns, Jörg— Fischer, Erik Frisk, Frank, Kasper B. Graversen, Alexandre Guimond, Cyril Goutte, Greg Gamble, Neil Hammond, Rasmus Borup Hansen, Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones, Johannes-Maria Kaltenbach, Michael Koundouros, Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo, Maik Lehradt, Alexander Mai, Martin Maechler, Aleksandar S Milosevic, Claus Malten, Kevin Van Maren, Lenimar Nunes de Andrade, Hubert Partl, John Refling, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Hanspeter Schmid, Craig Schlenter, Christopher Sawtell, Geoffrey Swindale, Josef Tkadlec, Didier Verna, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccone, and Mikhail Zotov.

Preface

LaTeX [1] is a typesetting system which is very suitable for producing scientific and mathematical documents of high typographical quality. The system is also suitable for producing all sorts of other documents, from simple letters to complete books. LaTeX uses TeX [2] as its formatting engine.

This short introduction describes LaTeX 2$_\varepsilon$ and should be sufficient for most applications of LaTeX. Refer to [1, 3] for a complete description of the LaTeX system.

LaTeX is available for most computers, from the PC and Mac to large UNIX and VMS systems. On many university computer clusters, you will find that a LaTeX installation is available, ready to use. Information on how to access the local LaTeX installation should be provided in the *Local Guide* [4]. If you have problems getting started, ask the person who gave you this booklet. The scope of this document is *not* to tell you how to install and set up a LaTeX system, but to teach you how to write your documents so that they can be processed by LaTeX.
This Introduction is split into 5 chapters:

Chapter 1 tells you about the basic structure of LaTeX 2$_\varepsilon$ documents. You will also learn a bit about the history of LaTeX. After reading this chapter, you should have a rough picture of LaTeX. The picture will only be a framework, but it will enable you to integrate the information provided in the other chapters into the big picture.

Chapter 2 goes into the details of typesetting your documents. It explains most of the essential LaTeX commands and environments. After reading this chapter, you will be able to write your first documents.

Chapter 3 explains how to typeset formulae with LaTeX. Again, a lot of examples help you to understand how to use one of LaTeX's main strengths. At the end of this chapter, you will find tables, listing all the mathematical symbols available in LaTeX.

Chapter 4 explains index and bibliography generation, inclusion of EPS graphics, and some other useful extensions.

Chapter 5 contains some potentially dangerous information about how to make alterations to the standard document layout produced by LaTeX. It will tell you how to change things such that the beautiful output of LaTeX begins looking quite bad.

It is important to read the chapters in sequential order. The book is not that big after all. Make sure to carefully read the examples, because a great part of the information is contained in the various examples you will find all throughout the book.

If you need to get hold of any LaTeX related material, have a look at one of the Comprehensive TeX Archive Network (`CTAN`) sites. The homesite is at `http://www.ctan.org`. All packages can also be retrieved from the ftp archive `ftp://www.ctan.org` and it's various mirror sites all over the world. They can be found e.g. at `ftp://ctan.tug.org` (US), `ftp://ftp.dante.de` (Germany), `ftp://ftp.tex.ac.uk` (UK). If you are not in one of these countries, choose the archive closest to you.

You will find other references to CTAN throughout the book. Especially pointers to software and documents you might want to download. Instead of writing down complete urls, I just wrote `CTAN:` followed by whatever location within the CTAN tree you should go to.

If you want to run LaTeX on your own computer, take a look at what is available from `CTAN:/tex-archive/systems`.

If you have ideas for something to be added, removed or altered in this document, please let me know. I am especially interested in feedback from LaTeX novices about which bits of this intro are easy to understand and which could be explained better.

Tobias Oetiker `<oetiker@ee.ethz.ch>`

Department of Electrical Engineering,
Swiss Federal Institute of Technology

The current version of this document is available on
`CTAN:/tex-archive/info/lshort`

Contents

Thank you! iii

Preface v

1 Things You Need to Know 1
1.1 The Name of the Game . . . 1
1.1.1 TeX . . . 1
1.1.2 LaTeX . . . 1
1.2 Basics . . . 2
1.2.1 Author, Book Designer, and Typesetter . . . 2
1.2.2 Layout Design . . . 3
1.2.3 Advantages and Disadvantages . . . 3
1.3 LaTeX Input Files . . . 5
1.3.1 Spaces . . . 5
1.3.2 Special Characters . . . 5
1.3.3 LaTeX Commands . . . 6
1.3.4 Comments . . . 6
1.4 Input File Structure . . . 7
1.5 A Typical Commandline Session . . . 9
1.6 The Layout of the Document . . . 9
1.6.1 Document Classes . . . 9
1.6.2 Packages . . . 10
1.7 Files you might encounter . . . 12
1.7.1 Page Styles . . . 14
1.8 Big Projects . . . 14

2 Typesetting Text 17
2.1 The Structure of Text and Language . . . 17
2.2 Linebreaking and Pagebreaking . . . 19
2.2.1 Justified Paragraphs . . . 19
2.2.2 Hyphenation . . . 20
2.3 Ready made Strings . . . 21
2.4 Special Characters and Symbols . . . 21

2.4.1 Quotation Marks 21
2.4.2 Dashes and Hyphens 22
2.4.3 Tilde ($\sim$) 22
2.4.4 Degree Symbol ($\circ$) 22
2.4.5 Ellipsis (. . .) 23
2.4.6 Ligatures 23
2.4.7 Accents and Special Characters 23
2.5 International Language Support 24
2.5.1 Support for German 26
2.6 The Space between Words 26
2.7 Titles, Chapters, and Sections 27
2.8 Cross References 29
2.9 Footnotes 29
2.10 Emphasized Words 30
2.11 Environments 30
2.11.1 Itemize, Enumerate, and Description 31
2.11.2 Flushleft, Flushright, and Center 31
2.11.3 Quote, Quotation, and Verse 32
2.11.4 Printing Verbatim 32
2.11.5 Tabular 33
2.12 Floating Bodies 35
2.13 Protecting fragile commands 38

3 Typesetting Mathematical Formulae 39
3.1 General 39
3.2 Grouping in Math Mode 41
3.3 Building Blocks of a Mathematical Formula 41
3.4 Math Spacing 45
3.5 Vertically Aligned Material 46
3.6 Phantom 48
3.7 Math Font Size 48
3.8 Theorems, Laws, ... 49
3.9 Bold symbols 50
3.10 List of Mathematical Symbols 52

4 Specialities 59
4.1 Including EPS Graphics 59
4.2 Bibliography 61
4.3 Indexing 62
4.4 Fancy Headers 63
4.5 The Verbatim Package 64
4.6 Downloading and Installing LaTeX Packages 65

5 Customising LaTeX **67**
5.1 New Commands, Environments and Packages 67
5.1.1 New Commands 68
5.1.2 New Environments 69
5.1.3 Your own Package 69
5.2 Fonts and Sizes 70
5.2.1 Font changing Commands 70
5.2.2 Danger, Will Robinson, Danger 73
5.2.3 Advice 74
5.3 Spacing 74
5.3.1 Line Spacing 74
5.3.2 Paragraph Formatting 74
5.3.3 Horizontal Space 75
5.3.4 Vertical Space 76
5.4 Page Layout 76
5.5 More fun with lengths 78
5.6 Boxes 79
5.7 Rules and Struts 81

Bibliography **83**

Index **85**

List of Figures

1.1 Components of a TEX System. 2
1.2 A Minimal LATEX File. 8
1.3 Example of a Realistic Journal Article. 8

4.1 Example fancyhdr Setup. 64

5.1 Example Package. 70
5.2 Page Layout Parameters. 77

List of Tables

1.1 Document Classes. 10
1.2 Document Class Options. 11
1.3 Some of the Packages Distributed with LaTeX. 12
1.4 The Predefined Page Styles of LaTeX. 14

2.1 Accents and Special Characters. 24
2.2 German Special Characters. 26
2.3 Float Placing Permissions. 36

3.1 Math Mode Accents. 52
3.2 Lowercase Greek Letters. 52
3.3 Uppercase Greek Letters. 52
3.4 Binary Relations. 53
3.5 Binary Operators. 53
3.6 BIG Operators. 54
3.7 Arrows. 54
3.8 Delimiters. 54
3.9 Large Delimiters. 54
3.10 Miscellaneous Symbols. 55
3.11 Non-Mathematical Symbols. 55
3.12 AMS Delimiters. 55
3.13 AMS Greek and Hebrew. 55
3.14 AMS Binary Relations. 56
3.15 AMS Arrows. 56
3.16 AMS Negated Binary Relations and Arrows. 57
3.17 AMS Binary Operators. 57
3.18 AMS Miscellaneous. 58
3.19 Math Alphabets. 58

4.1 Key Names for graphicx Package. 60
4.2 Index Key Syntax Examples. 63

5.1 Fonts. 71
5.2 Font Sizes. 71
5.3 Absolute Point Sizes in Standard Classes. 72

5.4 Math Fonts. 72
5.5 TEX Units. 76

Chapter 1

Things You Need to Know

In the first part of this chapter, you will get a short overview about the philosophy and history of LaTeX 2_ε. The second part of the chapter focuses on the basic structures of a LaTeX document. After reading this chapter, you should have a rough knowledge of how LaTeX works. When reading on, this will help you to integrate all the new information into the big picture.

1.1 The Name of the Game

1.1.1 TeX

TeX is a computer program created by Donald E. Knuth [2]. It is aimed at typesetting text and mathematical formulae. Knuth started writing the TeX typesetting engine in 1977 to explore the potential of the digital printing equipment that was beginning to infiltrate the publishing industry at that time, especially in the hope that he could reverse the trend of deteriorating typographical quality that he saw affecting his own books and articles. TeX as we use it today was released in 1982, with some slight enhancements added in 1989 to better support 8-bit characters and multiple languages. TeX is renowned for being extremely stable, for running on many different kinds of computers, and for being virtually bug free. The version number of TeX is converging to π and is now at 3.14159.

TeX is pronounced "Tech," with a "ch" as in the German word "Ach" or in the Scottish "Loch." In an ASCII environment, TeX becomes `TeX`.

1.1.2 LaTeX

LaTeX is a macro package which enables authors to typeset and print their work at the highest typographical quality, using a predefined, professional layout. LaTeX was originally written by Leslie Lamport [1]. It uses the TeX formatter as its typesetting engine.

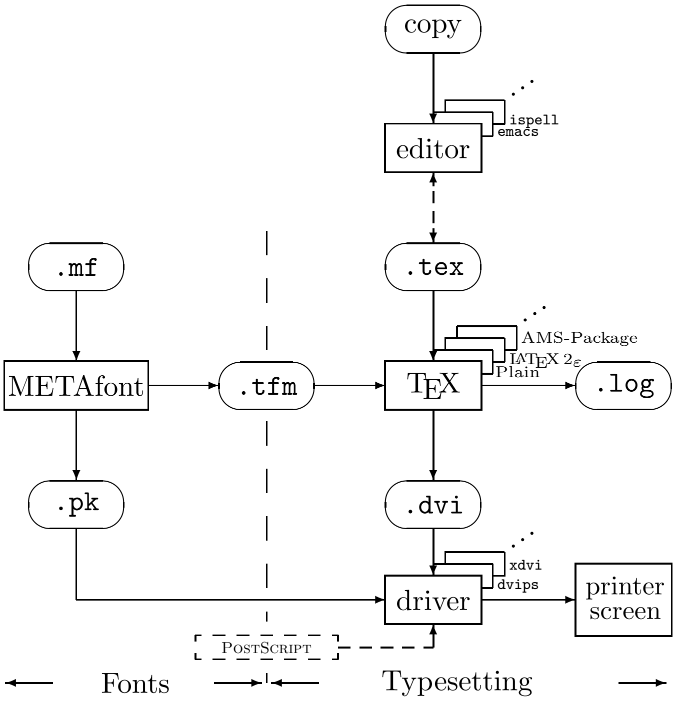


Figure 1.1: Components of a TEX System.

In 1994 the LaTeX package was updated by the LaTeX3 team, led by Frank Mittelbach, to include some long-requested improvements, and to reunify all the patched versions which had cropped up since the release of LaTeX 2.09 some years earlier. To distinguish the new version from the old, it is called LaTeX 2_ε. This documentation deals with LaTeX 2_ε.

LaTeX is pronounced "Lay-tech" or "Lah-tech." If you refer to LaTeX in an `ASCII` environment, you type `LaTeX`. LaTeX 2_ε is pronounced "Lay-tech two e" and typed `LaTeX2e`.

Figure 1.1 above shows how TeX and LaTeX 2_ε work together. This figure is taken from `wots.tex` by Kees van der Laan.

1.2 Basics

1.2.1 Author, Book Designer, and Typesetter

To publish something, authors give their typed manuscript to a publishing company. One of their book designers then decides the layout of the document (column width, fonts, space before and after headings, ...). The book designer writes his instructions into the manuscript and then gives it to a typesetter, who typesets the book according to these instructions.

A human book designer tries to find out what the author had in mind while writing the manuscript. He decides on chapter headings, citations, examples, formulae, etc. based on his professional knowledge and from the contents of the manuscript.

In a LaTeX environment, LaTeX takes the role of the book designer and uses TeX as its typesetter. But LaTeX is "only" a program and therefore needs more guidance. The author has to provide additional information which describes the logical structure of his work. This information is written into the text as "LaTeX commands."

This is quite different from the WYSIWYG[1] approach which most modern word processors such as *MS Word* or *Corel WordPerfect* take. With these applications, authors specify the document layout interactively while typing text into the computer. All along the way, they can see on the screen how the final work will look when it is printed.

When using LaTeX it is normally not possible to see the final output while typing the text. But the final output can be previewed on the screen after processing the file with LaTeX. Then corrections can be made before actually sending the document to the printer.

1.2.2 Layout Design

Typographical design is a craft. Unskilled authors often commit serious formatting errors by assuming that book design is mostly a question of aesthetics—"If a document looks good artistically, it is well designed." But as a document has to be read and not hung up in a picture gallery, the readability and understandability is of much greater importance than the beautiful look of it. Examples:

- The font size and the numbering of headings have to be chosen to make the structure of chapters and sections clear to the reader.
- The line length has to be short enough to not strain the eyes of the reader, while long enough to fill the page beautifully.

With WYSIWYG systems, authors often generate aesthetically pleasing documents with very little or inconsistent structure. LaTeX prevents such formatting errors by forcing the author to declare the *logical* structure of his document. LaTeX then chooses the most suitable layout.

1.2.3 Advantages and Disadvantages

When People from the WYSIWYG world meet people who use LaTeX, they often discuss "the advantages of LaTeX over a normal word processor" or the

[1] What you see is what you get.

opposite. The best thing you can do when such a discussion starts is to keep a low profile, since such discussions often get out of hand. But sometimes you cannot escape ...

So here is some ammunition. The main advantages of LaTeX over normal word processors are the following:

- Professionally crafted layouts are available, which make a document really look as if "printed."
- The typesetting of mathematical formulae is supported in a convenient way.
- The user only needs to learn a few easy-to-understand commands which specify the logical structure of a document. They almost never need to tinker with the actual layout of the document.
- Even complex structures such as footnotes, references, table of contents, and bibliographies can be generated easily.
- Free add-on packages exist for many typographical tasks not directly supported by basic LaTeX. For example, packages are available to include PostScript graphics or to typeset bibliographies conforming to exact standards. Many of these add-on packages are described in *The LaTeX Companion* [3].
- LaTeX encourages authors to write well-structured texts, because this is how LaTeX works—by specifying structure.
- TeX, the formatting engine of LaTeX 2$_\varepsilon$, is highly portable and free. Therefore the system runs on almost any hardware platform available.

LaTeX also has some disadvantages, and I guess it's a bit difficult for me to find any sensible ones, though I am sure other people can tell you hundreds `;-)`

- LaTeX does not work well for people who have sold their souls ...
- Although some parameters can be adjusted within a predefined document layout, the design of a whole new layout is difficult and takes a lot of time.[2]
- It is very hard to write unstructured and disorganized documents.
- Your hamster might, despite some encouraging first steps, never be able to fully grasp the concept of Logical Markup.

[2] Rumour says that this is one of the key elements which will be addressed in the upcoming LaTeX3 system.

1.3 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 tell LaTeX how to typeset the text.

1.3.1 Spaces

"Whitespace" characters such as blank or tab are treated uniformly as "space" by LaTeX. *Several consecutive* whitespace characters are treated as *one* "space". Whitespace at the start of a line is generally ignored, and a single linebreak is treated as "whitespace".

An empty line between two lines of text defines the end of a paragraph. *Several* empty lines are treated the same as *one* empty line. The text below is an example. On the left hand side is the text from the input file, and on the right hand side is the formatted output.

```
It does not matter whether you
enter one or several     spaces
after a word.

An empty line starts a new
paragraph.
```

It does not matter whether you enter one or several spaces after a word.

An empty line starts a new paragraph.

1.3.2 Special Characters

The following symbols are reserved characters that either have a special meaning under LaTeX or are not available in all the fonts. If you enter them directly in your text, they will normally not print, but rather coerce LaTeX to do things you did not intend.

```
# $ % ^ & _ { } ~ \
```

As you will see, these characters can be used in your documents all the same by adding a prefix backslash:

```
\# \$ \% \^{} \& \_ \{ \} \~{}
```

$ % ^ & _ { } ~

The other symbols and many more can be printed with special commands in mathematical formulae or as accents. The backslash character \ can *not* be entered by adding another backslash in front of it (\\), this sequence is used for linebreaking.[3]

[3] Try the `$\backslash$` command instead. It produces a '\'.

1.3.3 LaTeX Commands

LaTeX commands are case sensitive and take one of the following two formats:

- They start with a backslash `\` and then have a name consisting of letters only. Command names are terminated by a space, a number or any other 'non-letter'.
- They consist of a backslash and exactly one special character.

LaTeX ignores whitespace after commands. If you want to get a space after a command, you have to put either `{}` and a blank or a special spacing command after the command name. The `{}` stops LaTeX from eating up all the space after the command name.

```
I read that Knuth divides the
people working with \TeX{} into
\TeX{}nicians and \TeX perts.\\
Today is \today.
```

I read that Knuth divides the people working with TeX into TeXnicians and TeXperts.
Today is 9th August 2001.

Some commands need a parameter which has to be given between curly braces `{ }` after the command name. Some commands support optional parameters which are added after the command name in square brackets `[ ]`. The next examples use some LaTeX commands. Don't worry about them, they will be explained later.

```
You can \textsl{lean} on me!
```

You can *lean* on me!

```
Please, start a new line
right here!\newline
Thank you!
```

Please, start a new line right here!
Thank you!

1.3.4 Comments

When LaTeX encounters a `%` character while processing an input file, it ignores the rest of the present line, the linebreak, and all whitespace at the beginning of the next line.

This can be used to write notes into the input file, which will not show up in the printed version.

```
This is an % stupid
% Better: instructive <----
example: Supercal%
              ifragilist%
    icexpialidocious
```

This is an example: Supercalifragilisticexpialidocious

The % character can also be used to split long input lines where no whitespace or linebreaks are allowed.

For longer comments you should use the `comment` environment provided by the verbatim package. This means, to use the `comment` environment you have to add the commend `\usepackage{verbatim}` to the preamble of your document.

```
This is another
\begin{comment}
rather stupid,
but helpful
\end{comment}
example for embedding
comments in your document.
```

This is another example for embedding comments in your document.

Note that this won't work inside complex environments like math for example.

1.4 Input File Structure

When LaTeX 2_ε processes an input file, it expects it to follow a certain structure. Thus every input file must start with the command

```
\documentclass{...}
```

This specifies what sort of document you intend to write. After that, you can include commands which influence the style of the whole document, or you can load packages which add new features to the LaTeX system. To load such a package you use the command

```
\usepackage{...}
```

When all the setup work is done,[4] 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

```
\end{document}
```

command, which tells LaTeX to call it a day. Anything which follows this command will be ignored by LaTeX.

Figure 1.2 shows the contents of a minimal LaTeX 2_ε file. A slightly more complicated input file is given in Figure 1.3.

[4]The area between `\documentclass` and `\begin{document}` is called *preamble*.

```
\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}
```

Figure 1.2: A Minimal LaTeX File.

```
\documentclass[a4paper,11pt]{article}
% define the title
\author{H.~Partl}
\title{Minimalism}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Start}
Well, and here begins my lovely article.
\section{End}
\ldots{} and here it ends.
\end{document}
```

Figure 1.3: Example of a Realistic Journal Article.

1.5 A Typical Commandline Session

I bet you must be dying to try out the neat small LaTeX input file shown on page 8. Here is some help: LaTeX itself comes without a GUI or fancy buttons to press. It is just a program which crunches away on your input file. Some LaTeX installations feature a graphical front end where you can click LaTeX into compiling your input file. But Real Men don't Click, so here is how to coax LaTeX into compiling your input file on a text based system. Please note, this description assumes that a working LaTeX installation already sitts on your computer.

1. Edit/Create your LaTeX input file. This file must be plain ASCII text. On Unix all the editors will create just that. On windows you might want to make sure that you save the file in ASCII or *Plain Text* format. When picking a name for your file, make sure it bears the extention `.tex`.

2. Run LaTeX on your input file. If successful you will end up with a `.dvi` file.

   ```
   latex foo.tex
   ```

3. Now you may view the DVI file.

   ```
   xdvi foo.dvi
   ```

   or convert it to PS

   ```
   dvips -Pcmz foo.dvi -o foo.ps
   ```

   `xdvi` and `dvips` are open-source tools for handling `.dvi` files. The first displays them on screen within the X11 environment and the other creates a PostScript file for printing. If you are not working on a Unix system, other means for handling the `.dvi` files may be provided.

1.6 The Layout of the Document

1.6.1 Document Classes

The first information LaTeX needs to know when processing an input file is the type of document the author wants to create. This is specified with the `\documentclass` command.

`\documentclass[`*options*`]{`*class*`}`

Here *class* specifies the type of document to be created. Table 1.1 lists the document classes explained in this introduction. The LaTeX 2_ε distribution provides additional classes for other documents, including letters and slides. The *options* parameter customises the behaviour of the document class. The options have to be separated by commas. The most common options for the standard document classes are listed in Table 1.2.

Example: An input file for a LaTeX document could start with the line

```
\documentclass[11pt,twoside,a4paper]{article}
```

which instructs LaTeX to typeset the document as an *article* with a base font size of *eleven points*, and to produce a layout suitable for *double sided* printing on *A4 paper*.

1.6.2 Packages

While writing your document, you will probably find that there are some areas where basic LaTeX cannot solve your problem. If you want to include graphics, coloured text or source code from a file into your document, you need to enhance the capabilities of LaTeX. Such enhancements are called packages. Packages are activated with the

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

command where *package* is the name of the package and *options* is a list of keywords which trigger special features in the package. Some packages come with the LaTeX 2_ε base distribution (See Table 1.3). Others are provided separately. You may find more information on the packages installed at your site in your *Local Guide* [4]. The prime source for information about LaTeX packages is *The LaTeX Companion* [3]. It contains descriptions of hundreds

Table 1.1: Document Classes.

`article` for articles in scientific journals, presentations, short reports, program documentation, invitations, ...

`report` for longer reports containing several chapters, small books, PhD theses, ...

`book` for real books

`slides` for slides. The class uses big sans serif letters. You might want to consider using FoilTeX[a] instead.

[a] `CTAN:/tex-archive/macros/latex/contrib/supported/foiltex`

Table 1.2: Document Class Options.

`10pt`, `11pt`, `12pt` Sets the size of the main font in the document. If no option is specified, `10pt` is assumed.

`a4paper`, `letterpaper`, ... Defines the paper size. The default size is `letterpaper`. Besides that, `a5paper`, `b5paper`, `executivepaper`, and `legalpaper` can be specified.

`fleqn` Typesets displayed formulae left-aligned instead of centred.

`leqno` Places the numbering of formulae on the left hand side instead of the right.

`titlepage`, `notitlepage` Specifies whether a new page should be started after the document title or not. The `article` class does not start a new page by default, while `report` and `book` do.

`onecolumn`, `twocolumn` Instructs LaTeX to typeset the document in one columntwo columns.

`twoside, oneside` Specifies whether double or single sided output should be generated. The classes `article` and `report` are single sided and the `book` class is double sided by default. Note that this option concerns the style of the document only. The option `twoside` does *not* tell the printer you use that it should actually make a two-sided printout.

`openright, openany` Makes chapters begin either only on right hand pages or on the next page available. This does not work with the `article` class, as it does not know about chapters. The `report` class by default starts chapters on the next page available and the `book` class starts them on right hand pages.

of packages along with information of how to write your own extensions to LaTeX 2_ε.

Table 1.3: Some of the Packages Distributed with LaTeX.

doc Allows the documentation of LaTeX programs.
Described in `doc.dtx`[a] and in *The LaTeX Companion* [3].

exscale Provides scaled versions of the math extension font.
Described in `ltexscale.dtx`.

fontenc Specifies which font encoding LaTeX should use.
Described in `ltoutenc.dtx`.

ifthen Provides commands of the form
'if. . . then do. . . otherwise do. . . .'
Described in `ifthen.dtx` and *The LaTeX Companion* [3].

latexsym To access the LaTeX symbol font, you should use the `latexsym` package. Described in `latexsym.dtx` and in *The LaTeX Companion* [3].

makeidx Provides commands for producing indexes. Described in section 4.3 and in *The LaTeX Companion* [3].

syntonly Processes a document without typesetting it.

inputenc Allows the specification of an input encoding such as ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows or user-defined one.
Described in `inputenc.dtx`.

[a] This file should be installed on your system, and you should be able to get a `dvi` file by typing `latex doc.dtx` in any directory where you have write permission. The same is true for all the other files mentioned in this table.

1.7 Files you might encounter

When you work with LaTeX you will soon find yourself in a maze of files with various extensions and probably no clue. Below there is a list telling about the various file types you might encounter when working with TeX. Please note that this table does not claim to be a complete list of extensions, but if you find one missing which you think is important, please drop a line.

.tex LaTeX or TeX input file. Can be compiled with `latex`.

.sty LaTeX Macro package. This is a file you can load into your LaTeX document using the `\usepackage` command.

.dtx Documented TeX. This is the main distribution format for LaTeX style files. If you process a .dtx file you get documented macro code of the LaTeX package contained in the .dtx file.

.ins Is the installer for the files contained in the matching .dtx file. If you download a LaTeX package from the net, you will normally get a .dtx and a .ins file. Run LaTeX on the .ins file to unpack the .dtx file.

.cls Class files define what your document looks like. They are selected with the `\documentclass` command.

The following files are generated when you run LaTeX on your input file:

.dvi Device Independent file. This is the main result of a LaTeX compile run. You can look at its content with a DVI previewer program or you can send it to a printer with `dvips` or a similar application.

.log Gives a detailed account of what happened during the last compiler run.

.toc Stores all your section headers. It gets read in for the next compiler run and is used to produce the table of content.

.lof This is like .toc but for the list of figures.

.lot And again the same for the list of tables.

.aux Another file which transports information from one compiler run to the next. Among other things, the .aux file is used to store information associated with crossreferences.

.idx If your document contains an index. LaTeX stores all the words which go into the index in this file. Process this file with `makeindex`. Refer to section 4.3 on page 62 for more information on indexing.

.ind Is the processed .idx file, ready for inclusion into your document on the next compile cycle.

.ilg Logfile telling about what `makeindex` did.

1.7.1 Page Styles

LaTeX supports three predefined header/footer combinations—so-called page styles. The *style* parameter of the

`\pagestyle{`*style*`}`

command defines which one to use. Table 1.4 lists the predefined page styles.

Table 1.4: The Predefined Page Styles of LaTeX.

`plain` prints the page numbers on the bottom of the page, in the middle of the footer. This is the default page style.

`headings` prints the current chapter heading and the page number in the header on each page, while the footer remains empty. (This is the style used in this document)

`empty` sets both the header and the footer to be empty.

It is possible to change the page style of the current page with the command

`\thispagestyle{`*style*`}`

A description how to create your own headers and footers can be found in *The LaTeX Companion* [3] and in section 4.4 on page 63.

1.8 Big Projects

When working on big documents, you might want to split the input file into several parts. LaTeX has two commands which help you to do that.

`\include{`*filename*`}`

you can use this command in the document body to insert the contents of another file named *filename.tex*. Note that LaTeX will start a new page before processing the material input from *filename.tex*.

The second command can be used in the preamble. It allows you to instruct LaTeX to only input some of the `\include`d files.

`\includeonly{`*filename*`,`*filename*`,…}`

After this command is executed in the preamble of the document, only `\include` commands for the filenames which are listed in the argument of the `\includeonly` command will be executed. Note that there must be no spaces between the filenames and the commas.

The `\include` command starts typesetting the included text on a new page. This is helpful when you use `\includeonly`, because the pagebreaks will not move, even when some included files are omitted. Sometimes this might not be desirable. In this case, you can use the

`\input{`*filename*`}`

command. It simply includes the file specified. No flashy suits, no strings attached.

To make LaTeX quickly check your document you can use the syntonly package. This makes LaTeX skim through your document only checking for proper syntax and usage of the commands, but doesn't produce any (DVI) output. As LaTeX runs faster in this mode you may save yourself valuable time. Usage is very simple:

```
\usepackage{syntonly}
\syntaxonly
```

When you want to produce pages, just comment out the second line (by adding a percent sign).

Chapter 2

Typesetting Text

After reading the previous chapter, you should know about the basic stuff of which a LaTeX 2_ε document is made. In this chapter I will fill in the remaining structure you will need to know in order to produce real world material.

2.1 The Structure of Text and Language

The main point of writing a text (some modern DAAC[1] literature excluded), is to convey ideas, information, or knowledge to the reader. The reader will understand the text better if these ideas are well-structured, and will see and feel this structure much better if the typographical form reflects the logical and semantical structure of the content.

LaTeX is different from other typesetting systems in that you just have to tell it the logical and semantical structure of a text. It then derives the typographical form of the text according to the "rules" given in the document class file and in various style files.

The most important text unit in LaTeX (and in typography) is the paragraph. We call it "text unit" because a paragraph is the typographical form which should reflect one coherent thought, or one idea. You will learn in the following sections, how you can force linebreaks with e.g. \\ and paragraph breaks with e.g. leaving an empty line in the source code. Therefore, if a new thought begins, a new paragraph should begin, and if not, only linebreaks should be used. If in doubt about paragraph breaks, think about your text as a conveyor of ideas and thoughts. If you have a paragraph break, but the old thought continues, it should be removed. If some totally new line of thought occurs in the same paragraph, then it should be broken.

Most people completely underestimate the importance of well-placed paragraph breaks. Many people do not even know what the meaning of

[1] Different At All Cost, a translation of the Swiss German UVA (Um's Verrecken Anders).

a paragraph break is, or, especially in LaTeX, introduce paragraph breaks without knowing it. The latter mistake is especially easy to make if equations are used in the text. Look at the following examples, and figure out why sometimes empty lines (paragraph breaks) are used before and after the equation, and sometimes not. (If you don't yet understand all commands well enough to understand these examples, please read this and the following chapter, and then read this section again.)

```
% Example 1
\ldots when Einstein introduced his formula
\begin{equation}
  e = m \cdot c^2 \; ,
\end{equation}
which is at the same time the most widely known
and the least well understood physical formula.


% Example 2
\ldots from which follows Kirchoff's current law:
\begin{equation}
  \sum_{k=1}^{n} I_k = 0 \; .
\end{equation}

Kirchhoff's voltage law can be derived \ldots


% Example 3
\ldots which has several advantages.

\begin{equation}
  I_D = I_F - I_R
\end{equation}
is the core of a very different transistor model. \ldots
```

The next smaller text unit is a sentence. In English texts, there is a larger space after a period which ends a sentence than after one which ends an abbreviation. LaTeX tries to figure out which one you wanted to have. If LaTeX gets it wrong, you must tell it what you want. This is explained later in this chapter.

The structuring of text even extends to parts of sentences. Most languages have very complicated punctuation rules, but in many languages (including German and English), you will get almost every comma right if you remember what it represents: a short stop in the flow of language. If you are not sure about where to put a comma, read the sentence aloud, and take a short breath at every comma. If this feels awkward at some place, delete that comma, if you feel the urge to breathe (or make a short stop) at some other place, insert a comma.

Finally, the paragraphs of a text should also be structured logically at a higher level, by putting them into chapters, sections, subsections, and so on. However, the typographical effect of writing e.g. `\section{The Structure of Text and Language}` is so obvious that it is almost self-evident how these high-level structures should be used.

2.2 Linebreaking and Pagebreaking

2.2.1 Justified Paragraphs

Often books are typeset with each line having the same length. LaTeX inserts the necessary linebreaks and spaces between words by optimizing the contents of a whole paragraph. If necessary, it also hyphenates words that would not fit comfortably on a line. How the paragraphs are typeset depends on the document class. Normally the first line of a paragraph is indented, and there is no additional space between two paragraphs. Refer to section 5.3.2 for more information.

In special cases it might be necessary to order LaTeX to break a line:

`\\` or `\newline`

starts a new line without starting a new paragraph.

`\\*`

additionally prohibits a pagebreak after the forced linebreak.

`\newpage`

starts a new page.

`\linebreak[`*n*`]`, `\nolinebreak[`*n*`]`, `\pagebreak[`*n*`]` and `\nopagebreak[`*n*`]`

do what their names say. They enable the author to influence their actions with the optional argument n. It can be set to a number between zero to four. By setting n to a value below 4 you leave LaTeX the option of ignoring your command if the result would look very bad. Do not confuse these "break" commands with the "new" commands. Even when you give a "break" command, LaTeX still tries to even out the right border of the page and the total length of the page as described in the next section. If you really want to start a "new line", then use the corresponding command. Guess its name!

LaTeX always tries to produce the best linebreaks possible. If it cannot find a way to break the lines in a manner which meets its high standards, it lets one line stick out on the right of the paragraph. LaTeX then complains ("overfull hbox") while processing the input file. This happens most often when LaTeX cannot find a suitable place to hyphenate a word.[2] You can instruct LaTeX to lower its standards a little by giving the `\sloppy` command. It prevents such over-long lines by increasing the inter-word spacing — even if the final output is not optimal. In this case a warning ("underfull hbox") is given to the user. In most such cases the result doesn't look very good. The command `\fussy` brings LaTeX back to its default behaviour.

2.2.2 Hyphenation

LaTeX hyphenates words whenever necessary. If the hyphenation algorithm does not find the correct hyphenation points, you can remedy the situation by using the following commands to tell TeX about the exception.

The command

`\hyphenation{`*word list*`}`

causes the words listed in the argument to be hyphenated only at the points marked by "`-`". The argument of the command should only contain words built from normal letters or rather signes which are regarded as normal letters in the active context. The hyphenation hints are stored for the language which is active when the hyphenation command occurs. This means that if you place a hyphenation command into the preamble of your document it will influence the english language hyphenation. If you place the command after the `\begin{document}` and you are using some package for national language support like babel, then the hyphenation hints will be active in the language activated through babel.

The example below will allow "hyphenation" to be hyphenated as well as "Hyphenation", and it prevents "FORTRAN", "Fortran" and "fortran" from being hyphenated at all. No special characters or symbols are allowed in the argument.

Example:

```
\hyphenation{FORTRAN Hy-phen-a-tion}
```

[2]Although LaTeX gives you a warning when that happens (Overfull hbox) and displays the offending line, such lines are not always easy to find. If you use the option `draft` in the `\documentclass` command, these lines will be marked with a thick black line on the right margin.

The command `\-` inserts a discretionary hyphen into a word. This also becomes the only point hyphenation is allowed in this word. This command is especially useful for words containing special characters (e.g. accented characters), because LaTeX does not automatically hyphenate words containing special characters.

```
I think this is: su\-per\-cal\-%
i\-frag\-i\-lis\-tic\-ex\-pi\-%
al\-i\-do\-cious
```

I think this is: supercalifragilisticexpialido-
cious

Several words can be kept together on one line with the command

`\mbox{`*text*`}`

It causes its argument to be kept together under all circumstances.

```
My phone number will change soon.
It will be \mbox{0116 291 2319}.

The parameter
\mbox{\emph{filename}} should
contain the name of the file.
```

My phone number will change soon. It will be 0116 291 2319.

The parameter *filename* should contain the name of the file.

`\fbox` is similar to mbox, but in addition there will be a visible box drawn around the content.

2.3 Ready made Strings

In some of the examples on the previous pages you have seen some very simple LaTeX commands for typesetting special text strings:

Command	Example	Description
`\today`	9th August 2001	Current date in the current language
`\TeX`	TeX	The name of your favorite typesetter
`\LaTeX`	LaTeX	The name of the Game
`\LaTeXe`	LaTeX 2_ε	The current incarnation of LaTeX

2.4 Special Characters and Symbols

2.4.1 Quotation Marks

You should *not* use the `"` for quotation marks as you would on a typewriter. In publishing there are special opening and closing quotation marks. In LaTeX, use two `` ` ``s (grace accent) for opening quotation marks and two `'`s (apostrophe) for closing quotation marks. For single quotes you use just one of each.

```
``Please press the `x' key.''
```

“Please press the ‘x’ key.”

2.4.2 Dashes and Hyphens

LaTeX knows four kinds of dashes. You can access three of them with different numbers of consecutive dashes. The fourth sign is actually not a dash at all: It is the mathematical minus sign:

```
daughter-in-law, X-rated\\
pages 13--67\\
yes---or no? \\
$0$, $1$ and $-1$
```

daughter-in-law, X-rated
pages 13–67
yes—or no?
0, 1 and -1

The names for these dashes are: ‘-’ hyphen, ‘–’ en-dash, ‘—’ em-dash and ‘$-$’ minus sign.

2.4.3 Tilde ($\sim$)

A character, often seen with web addresses is the tilde. To generate this in LaTeX you can use `\~` but the result: ˜ is not really what you want. Try this instead:

```
http://www.rich.edu/\~{}bush \\
http://www.clever.edu/$\sim$demo
```

http://www.rich.edu/˜bush
http://www.clever.edu/$\sim$demo

2.4.4 Degree Symbol ($\circ$)

How to print a degree symbol in LaTeX?

```
Its $-30\,^{\circ}\mathrm{C}$,
I will soon start to
super-conduct.
```

Its $-30\,^{\circ}\mathrm{C}$, I will soon start to super-conduct.

2.4.5 Ellipsis (. . .)

On a typewriter a comma or a period takes the same amount of space as any other letter. In book printing these characters occupy only a little space and are set very close to the preceding letter. Therefore you cannot enter 'ellipsis' by just typing three dots, as the spacing would be wrong. Besides that there is a special command for these dots. It is called

```
\ldots
```

```
Not like this ... but like this:\\
New York, Tokyo, Budapest, \ldots
```

Not like this ... but like this:
New York, Tokyo, Budapest, . . .

2.4.6 Ligatures

Some letter combinations are typeset not just by setting the different letters one after the other, but by actually using special symbols.

ﬀ ﬁ ﬂ ﬃ. . . instead of ff fi fl ffi . . .

These so-called ligatures can be prohibited by inserting an `\mbox{}` between the two letters in question. This might be necessary with words built from two words.

```
Not shelfful\\
but shelf\mbox{}ful
```

Not shelﬀul
but shelfful

2.4.7 Accents and Special Characters

LaTeX supports the use of accents and special characters from many languages. Table 2.1 shows all sorts of accents being applied to the letter o. Naturally other letters work too.

To place an accent on top of an i or a j, its dots have to be removed. This is accomplished by typing `\i` and `\j`.

```
H\^otel, na\"\i ve, \'el\`eve,\\
sm\o rrebr\o d, !`Se\~norita!,\\
Sch\"onbrunner Schlo\ss{}
Stra\ss e
```

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

2.5 International Language Support

If you need to write documents in languages other than English, there are two areas where LaTeX has to be configured appropriately:

1. All automatically generated text strings[3] have to be adapted to the new language. For many languages, these changes can be accomplished by using the babel package by Johannes Braams.

2. LaTeX needs to know the hyphenation rules for the new language. Getting hyphenation rules into LaTeX is a bit more tricky. It means rebuilding the format file with different hyphenation patterns enabled. Your *Local Guide* [4] should give more information on this.

If your system is already configured appropriately, you can activate the babel package by adding the command

```
\usepackage[language]{babel}
```

after the `\documentclass` command. The *language*s your system supports should also be listed in the Local Guide. Babel will automatically activate the apropriate hyphenation rules for the language you choose. If your LaTeX format does not support hyphenation in the language of your choice, babel will still work but it will disable hyphenation which has quite a negative effect on the visual appearance of the typeset document.

For some languages, babel also specifies new commands which simplify the input of special characters. The German language, for example, contains

[3] Table of Contents, List of Figures, . . .

Table 2.1: Accents and Special Characters.

ò	`` \`o ``	ó	`\'o`	ô	`\^o`	õ	`\~o`
ō	`\=o`	ȯ	`\.o`	ö	`\"o`	ç	`\c c`
ŏ	`\u o`	ǒ	`\v o`	ő	`\H o`	ǫ	`\c o`
ọ	`\d o`	o̱	`\b o`	o͡o	`\t oo`		
œ	`\oe`	Œ	`\OE`	æ	`\ae`	Æ	`\AE`
å	`\aa`	Å	`\AA`				
ø	`\o`	Ø	`\O`	ł	`\l`	Ł	`\L`
ı	`\i`	ȷ	`\j`	¡	`` !` ``	¿	`` ?` ``

a lot of umlauts (äöü). With babel, you can enter an ö by typing `"o` instead of `\"o`.

Some computer systems allow you to input special characters directly from the keyboard. LaTeX can handle such characters. Since the December 1994 release of LaTeX 2ε, support for several input encodings is included in the basic distribution of LaTeX 2ε. Check the inputenc package:

```
\usepackage[encoding]{inputenc}
```

When using this package, you should consider that other people might not be able to display your input files on their computer, because they use a different encoding. For example, the German umlaut ä on a PC is encoded as 132, but on some Unix systems using ISO-LATIN 1 it is encoded as 228. Therefore you should use this feature with care. The following encodings may come handy, depending on the type of system you are working on:

Operating system	encoding
Mac	applemac
Unix	latin1
Windows	ansinew
OS/2	cp850

Font encoding is a different matter. It defines at which position inside a TeX-font each letter is stored. The original Computer Modern TeX font does only contain the 128 characters of the old 7-bit ASCII character set. When accented characters are required, TeX creates them by combining a normal character with an accent. While the resulting output looks perfect, this approach stops the automatic hyphenation from working inside words containing accented characters.

Fortunately, most modern TeX distributions contain a copy of the EC fonts. These fonts look like the Computer Modern fonts, but contain special characters for most of the accented characters used in European languages. By using these fonts you can improve hyphenation in non-English documents. The EC fonts are activated by including the fontenc package in the preamble of your document.

```
\usepackage[T1]{fontenc}
```

2.5.1 Support for German

Some hints for those creating German documents with LaTeX. You can load German language support with the command:

```
\usepackage[german]{babel}
```

This enables German hyphenation, if you have configured your LaTeX system accordingly. It also changes all automatic text into German. Eg. “Chapter” becomes “Kapitel”. Further a set of new commands becomes available which allows you to write German input files more quickly. Check out table 2.2 for inspiration.

Table 2.2: German Special Characters.

`"a`	ä	`"s`	ß
`"‘`	„	`"’`	“
`"<`	«	`">`	»
`\dq`	"		

2.6 The Space between Words

To get a straight right margin in the output, LaTeX inserts varying amounts of space between the words. It inserts slightly more space at the end of a sentence, as this makes the text more readable. LaTeX assumes that sentences end with periods, question marks or exclamation marks. If a period follows an uppercase letter, this is not taken as a sentence ending, since periods after uppercase letters normally occur in abbreviations.

Any exception from these assumptions has to be specified by the author. A backslash in front of a space generates a space which will not be enlarged. A tilde ‘~’ character generates a space which cannot be enlarged and which additionally prohibits a linebreak. The command `\@` in front of a period specifies that this period terminates a sentence even when it follows an uppercase letter.

```
Mr.~Smith was happy to see her\\
cf.~Fig.~5\\
I like BASIC\@. What about you?
```

Mr. Smith was happy to see her
cf. Fig. 5
I like BASIC. What about you?

The additional space after periods can be disabled with the command

```
\frenchspacing
```

which tells LaTeX *not* to insert more space after a period than after ordinary character. This is very common in non-English languages, except bibliographies. If you use `\frenchspacing`, the command `\@` is not necessary.

2.7 Titles, Chapters, and Sections

To help the reader find his or her way through your work, you should divide it into chapters, sections, and subsections. LaTeX supports this with special commands which take the section title as their argument. It is up to you to use them in the correct order.

The following sectioning commands are available for the `article` class:

```
\section{...}           \paragraph{...}
\subsection{...}        \subparagraph{...}
\subsubsection{...}
```

You can use two additional sectioning commands for the `report` and the `book` class:

```
\part{...}              \chapter{...}
```

As the `article` class does not know about chapters, it is quite easy to add articles as chapters to a book. The spacing between sections, the numbering and the font size of the titles will be set automatically by LaTeX.

Two of the sectioning commands are a bit special:

- The `\part` command does not influence the numbering sequence of chapters.
- The `\appendix` command does not take an argument. It just changes the chapter numbering to letters.[4]

LaTeX creates a table of contents by taking the section headings and page numbers from the last compile cycle of the document. The command

```
\tableofcontents
```

expands to a table of contents at the place where it is issued. A new document has to be compiled ("LaTeXed") twice to get a correct table of contents. Sometimes it might be necessary to compile the document a third time. LaTeX will tell you when this is necessary.

[4]For the article style it changes the section numbering.

All sectioning commands listed above also exist as "starred" versions. A "starred" version of a command is built by adding a star `*` after the command name. They generate section headings which do not show up in the table of contents and which are not numbered. The command `\section{Help}`, for example, would become `\section*{Help}`.

Normally the section headings show up in the table of contents exactly as they are entered in the text. Sometimes this is not possible, because the heading is too long to fit into the table of contents. The entry for the table of contents can then be specified as an optional argument in front of the actual heading.

```
\chapter[Title for the table of contents]{A long
    and especially boring title, shown in the text}
```

The title of the whole document is generated by issuing a

`\maketitle`

command. The contents of the title have to be defined by the commands

`\title{...}`, `\author{...}` and optionally `\date{...}`

before calling `\maketitle`. In the argument of `\author`, you can supply several names separated by `\and` commands.

An example of some of the commands mentioned above can be found in Figure 1.3 on page 8.

Apart from the sectioning commands explained above, LaTeX 2_ε introduced three additional commands for use with the `book` class. They are useful for dividing your publication. The commands alter chapter headings and page numbering to work as you would expect it in a book:

`\frontmatter` should be the very first command after `\begin{document}`. It will switch page numbering to Roman numerals. It is common to use the starred sectioning commands (eg `\chapter*{Preface}`) for frontmatter as this stopps LaTeX from enumerating them.

`\mainmatter` comes after right befor the first chapter of the book. It turns on Arabic page numbering and restarts the page counter.

`\appendix` marks the start of additional material in your book. After this command chapters will be numbered with letters.

`\backmatter` should be inserted before the very last items in your book like the bibliography and the index. In the standard document classes, this has no visual effect.

2.8 Cross References

In books, reports and articles, there are often cross-references to figures, tables and special segments of text. LaTeX provides the following commands for cross referencing

`\label{`*marker*`}`, `\ref{`*marker*`}` and `\pageref{`*marker*`}`

where *marker* is an identifier chosen by the user. LaTeX replaces `\ref` by the number of the section, subsection, figure, table, or theorem after which the corresponding `\label` command was issued. `\pageref` prints the page number of the page where the `\label` command occurred.[5] Just as the section titles, the numbers from the previous run are used.

```
A reference to this subsection
\label{sec:this} looks like:
``see section~\ref{sec:this} on
page~\pageref{sec:this}.''
```

A reference to this subsection looks like: "see section 2.8 on page 29."

2.9 Footnotes

With the command

`\footnote{`*footnote text*`}`

a footnote is printed at the foot of the current page. Footnotes should always be put[6] after the word or sentence they refer to. Footnotes referring to a sentence or part of it should therefore be put after the comma or period.[7]

```
Footnotes\footnote{This is
  a footnote.} are often used
by people using \LaTeX.
```

Footnotes[a] are often used by people using LaTeX.

[a]This is a footnote.

[5]Note that these commands are not aware of what they refer to. `\label` just saves the last automatically generated number.

[6]"put" is one of the most common English words.

[7]Note, that footnotes are distracting the reader from the main body of your document. After all everybody reads the footnotes, we are a curious species. So why not just integrate everything you want to say into the body of the document.[8]

[8]A guidepost doesn't necessarily go where it's pointing to :-).

2.10 Emphasized Words

If a text is typed using a typewriter, `important words are emphasized by underlining them.`

`\underline{`*text*`}`

In printed books, however, words are emphasized by typesetting them in an *italic* font. LaTeX provides the command

`\emph{`*text*`}`

to emphasize text. What the command actually does with its argument depends on the context:

```
\emph{If you use
  emphasizing inside a piece
  of emphasized text, then
  \LaTeX{} uses the
  \emph{normal} font for
  emphasizing.}
```

If you use emphasizing inside a piece of emphasized text, then LaTeX uses the normal *font for emphasizing.*

Please note the difference between telling LaTeX to *emphasize* something and telling it to use a different *font*:

```
\textit{You can also
  \emph{emphasize} text if
  it is set in italics,}
\textsf{in a
  \emph{sans-serif} font,}
\texttt{or in
  \emph{typewriter} style.}
```

You can also emphasize *text if it is set in italics,* in a *sans-serif* font, or in *typewriter* style.

2.11 Environments

`\begin{`*environment*`}` *text* `\end{`*environment*`}`

Where *environment* is the name of the environment. Environments can be called several times within each other as long as the calling order is maintained.

```
\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}
```

In the following sections all important environments are explained.

2.11.1 Itemize, Enumerate, and Description

The `itemize` environment is suitable for simple lists, the `enumerate` environment for enumerated lists, and the `description` environment for descriptions.

```
\flushleft
\begin{enumerate}
\item You can mix the list
environments to your taste:
\begin{itemize}
\item But it might start to
look silly.
\item[-] With a dash.
\end{itemize}
\item Therefore remember:
\begin{description}
\item[Stupid] things will not
become smart because they are
in a list.
\item[Smart] things, though, can be
presented beautifully in a list.
\end{description}
\end{enumerate}
```

1. You can mix the list environments to your taste:
   - But it might start to look silly.
   - With a dash.
2. Therefore remember:

   Stupid things will not become smart because they are in a list.

   Smart things, though, can be presented beautifully in a list.

2.11.2 Flushleft, Flushright, and Center

The environments `flushleft` and `flushright` generate paragraphs which are either left- or right-aligned. The `center` environment generates centred text. If you do not issue \\ to specify linebreaks, LaTeX will automatically determine linebreaks.

```
\begin{flushleft}
This text is\\ left-aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushleft}
```

This text is
left-aligned. LaTeX is not trying to make
each line the same length.

```
\begin{flushright}
This text is right-\\aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushright}
```

This text is right-
aligned. LaTeX is not trying to make each
line the same length.

```
\begin{center}
At the centre\\of the earth
\end{center}
```

At the centre
of the earth

2.11.3 Quote, Quotation, and Verse

The `quote` environment is useful for quotes, important phrases and examples.

```
A typographical rule of thumb
for the line length is:
\begin{quote}
On average, no line should
be longer than 66 characters.
\end{quote}
This is why \LaTeX{} pages have
such large borders by default and
also why multicolumn print is
used in newspapers.
```

A typographical rule of thumb for the line length is:

> On average, no line should be longer than 66 characters.

This is why LaTeX pages have such large borders by default and also why multicolumn print is used in newspapers.

There are two similar environments: the `quotation` and the `verse` environments. The `quotation` environment is useful for longer quotes going over several paragraphs, because it does indent paragraphs. The `verse` environment is useful for poems where the line breaks are important. The lines are separated by issuing a `\\` at the end of a line and a empty line after each verse.

```
I know only one English poem by
heart. It is about Humpty Dumpty.
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\\
Humpty Dumpty had a great fall.\\
All the King's horses and all
the King's men\\
Couldn't put Humpty together
again.
\end{verse}
\end{flushleft}
```

I know only one English poem by heart. It is about Humpty Dumpty.

> Humpty Dumpty sat on a wall:
> Humpty Dumpty had a great
> fall.
> All the King's horses and all
> the King's men
> Couldn't put Humpty together
> again.

2.11.4 Printing Verbatim

Text which is enclosed between `\begin{verbatim}` and `\end{verbatim}` will be directly printed, as if it was typed on a typewriter, with all linebreaks and spaces, without any LaTeX command being executed.

Within a paragraph, similar behavior can be accessed with

`\verb+`*text*`+`

The `+` is just an example of a delimiter character. You can use any character except letters, `*` or space. Many LaTeX examples in this booklet are typeset with this command.

```
The \verb|\ldots| command \ldots

\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

The `\ldots` command …

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

```
\begin{verbatim*}
the starred version of
the      verbatim
environment emphasizes
the spaces   in the text
\end{verbatim*}
```

```
the␣starred␣version␣of
the␣␣␣␣␣␣verbatim
environment␣emphasizes
the␣spaces␣␣␣in␣the␣text
```

The `\verb` command can be used in a similar fashion with a star:

```
\verb*|like   this :-) |
```

```
like␣␣␣this␣:-)␣
```

The `verbatim` environment and the `\verb` command may not be used within parameters of other commands.

2.11.5 Tabular

The `tabular` environment can be used to typeset beautiful tables with optional horizontal and vertical lines. LaTeX determines the width of the columns automatically.

The *table spec* argument of the

`\begin{tabular}{`*table spec*`}`

command defines the format of the table. Use an `l` for a column of left-aligned text, `r` for right-aligned text, and `c` for centred text; `p{`*width*`}` for a column containing justified text with linebreaks, and `|` for a vertical line.

Within a `tabular` environment, `&` jumps to the next column, `\\` starts a new line and `\hline` inserts a horizontal line. You can add partial Lines by using the `\cline{j-i}` whereby j and i are the column numbers the line should extend over.

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binary \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal

```
\begin{tabular}{|p{4.7cm}|}
\hline
Welcome to Boxy's paragraph.
We sincerely hope you'll
all enjoy the show.\\
\hline
\end{tabular}
```

Welcome to Boxy's paragraph. We sincerely hope you'll all enjoy the show.

The column separator can be specified with the `@{...}` construct. This command kills the inter-column space and replaces it with whatever is between the curly braces. One common use for this command is explained below in the decimal alignment problem. Another possible application is to suppress leading space in a table with `@{}`.

```
\begin{tabular}{@{} l @{}}
\hline
no leading space\\
\hline
\end{tabular}
```

no leading space

```
\begin{tabular}{l}
\hline
leading space left and right\\
\hline
\end{tabular}
```

leading space left and right

Since there is no built-in way to align numeric columns to a decimal point,[9] we can "cheat" and do it by using two columns: a right-aligned integer and a left-aligned fraction. The `@{.}` command in the `\begin{tabular}` line replaces the normal inter-column spacing with just a ".", giving the appearance of a single, decimal-point-justified column. Don't forget to replace the decimal point in your numbers with a column separator (`&`)! A column

[9]If the 'tools' bundle is installed on your system, have a look at the dcolumn package.

label can be placed above our numeric "column" by using the `\multicolumn` command.

```
\begin{tabular}{c r @{.} l}
Pi expression       &
\multicolumn{2}{c}{Value} \\
\hline
$\pi$               & 3&1416  \\
$\pi^{\pi}$         & 36&46   \\
$(\pi^{\pi})^{\pi}$ & 80662&7 \\
\end{tabular}
```

Pi expression	Value
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Ene} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

Ene	
Mene	Muh!

Material typeset with the tabular environment always stays together on one page. If you want to typeset long tables you might want to have a look at the supertabular and the longtabular environments.

2.12 Floating Bodies

Today most publications contain a lot of figures and tables. These elements need special treatment, because they cannot be broken across pages. One method would be to start a new page every time a figure or a table is too large to fit on the present page. This approach would leave pages partially empty, which looks very bad.

The solution to this problem is to 'float' any figure or table which does not fit on the current page to a later page, while filling the current page with body text. LaTeX offers two environments for floating bodies; one for tables and one for figures. To take full advantage of these two environments it is important to understand approximately how LaTeX handles floats internally. Otherwise floats may become a major source of frustration, because LaTeX never puts them where you want them to be.

Let's first have a look at the commands LaTeX supplies for floats:

Any material enclosed in a `figure` or `table` environment will be treated as floating matter. Both float environments support an optional parameter

`\begin{figure}[`*placement specifier*`]` or `\begin{table}[`*placement specifier*`]`

called the *placement specifier*. This parameter is used to tell LaTeX about the locations to which the float is allowed to be moved. A *placement specifier* is constructed by building a string of *float-placing permissions*. See Table 2.3.

Note: The `0pt` and `1.05em` are TeX units. Read more on this in table 5.5 on page 76.

A table could be started with the following line e.g.

```
\begin{table}[!hbp]
```

The placement specifier `[!hbp]` allows LaTeX to place the table right here (`h`) or at the bottom (`b`) of some page or on a special floats page (`p`), and all this even if it does not look that good (`!`). If no placement specifier is given, the standard classes assume `[tbp]`.

LaTeX will place every float it encounters, according to the placement specifier supplied by the author. If a float cannot be placed on the current page it is deferred either to the *figures* or the *tables* queue[10]. When a new page is started, LaTeX first checks if it is possible to fill a special 'float' page with floats from the queues. If this is not possible, the first float on each queue is treated as if it had just occurred in the text: LaTeX tries again to place it according to its respective placement specifiers (except 'h' which is no longer possible). Any new floats occurring in the text get placed into the appropriate queues. LaTeX strictly maintains the original order of appearance for each type of float. That's why a figure which cannot be placed pushes all further figures to the end of the document. Therefore:

[10] These are fifo - 'first in first out' queues!

Table 2.3: Float Placing Permissions.

Spec	Permission to place the float ...
`h`	*here* at the very place in the text where it occurred. This is useful mainly for small floats.
`t`	at the *top* of a page
`b`	at the *bottom* of a page
`p`	on a special *page* containing only floats.
`!`	without considering most of the internal parameters[a] which could stop this float from being placed.

[a] Such as the maximum number of floats allowed on one page.

> If LaTeX is not placing the floats as you expected, it is often only one float jamming one of the two float queues.

While it is possible to give LaTeX single-location placement specifiers, this causes problems. If the float does not fit in the location specified, then it becomes stuck, blocking subsequent floats. In particular, you should never ever use the [h] option, it is so bad that in more recent versions of LaTeX, it is automatically replaced by [ht].

Having explained the difficult bit, there are some more things to mention about the `table` and `figure` environments. With the

`\caption{`*caption text*`}`

command, you can define a caption for the float. A running number and the string "Figure" or "Table" will be added by LaTeX.

The two commands

`\listoffigures` and `\listoftables`

operate analogously to the `\tableofcontents` command, printing a list of figures or tables, respectively. In these lists, the whole caption will be repeated. If you tend to use long captions, you must have a shorter version of the caption going into the lists. This is accomplished by entering the short version in brackets after the `\caption` command.

```
\caption[Short]{LLLLLoooooonnnnnggggg}
```

With `\label` and `\ref`, you can create a reference to a float within your text.

The following example draws a square and inserts it into the document. You could use this if you wanted to reserve space for images you are going to paste into the finished document.

```
Figure~\ref{white} is an example of Pop-Art.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Five by Five in Centimetres.} \label{white}
\end{figure}
```

In the example above, LaTeX will try *really hard* (`!`) to place the figure right *here* (`h`).[11] If this is not possible, it tries to place the figure at the *bottom* (`b`) of the page. Failing to place the figure on the current page, it determines whether it is possible to create a float page containing this

[11]assuming the figure queue is empty.

figure and maybe some tables from the tables queue. If there is not enough material for a special float page, LaTeX starts a new page, and once more treats the figure as if it had just occurred in the text.

Under certain circumstances it might be necessary to use the

`\clearpage` or even the `\cleardoublepage`

command. It orders LaTeX to immediately place all floats remaining in the queues and then start a new page. `\cleardoublepage` even goes to a new righthand page.

You will learn how to include PostScript drawings into your LaTeX 2_ε documents later in this introduction.

2.13 Protecting fragile commands

Text given as arguments of commands like `\caption` or `\section` may show up more than once in the document (e.g. in the table of contents as well as in the body of the document). Some commands fail when used in the argument of `\section`-like commands. These are called fragile commands. Fragile commands are for example `\footnote` or `\phantom`. What these fragile commands need to work, is protection (don't we all?). You can protect them by putting the `\protect` command in front of them.

`\protect` only refers to the command which follows right behind, not even to its arguments. In most cases a superfluous `\protect` won't hurt.

```
\section{I am considerate
      \protect\footnote{and protect my footnotes}}
```

Chapter 3

Typesetting Mathematical Formulae

Now you are ready! In this chapter, we will attack the main strength of TEX: mathematical typesetting. But be warned, this chapter only scratches the surface. While the things explained here are sufficient for many people, don't despair if you can't find a solution to your mathematical typesetting needs here. It is highly likely that your problem is addressed in $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX[1] or some other package.

3.1 General

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

[1] `CTAN:/tex-archive/macros/latex/required/amslatex`

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
```

$$\epsilon > 0 \tag{3.1}$$

From (3.1), 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}
```

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

There are differences between *math mode* and *text mode*. For example in *math mode*:

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}
```

$$\forall x \in \mathbf{R}: \qquad x^{2} \geq 0 \tag{3.2}$$

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

$$x^{2} \geq 0\qquad \textrm{for all }x\in\mathbf{R} \tag{3.3}$$

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}
```

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

3.2 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}
```

$$a^x+y \neq a^{x+y} \tag{3.4}$$

3.3 Building Blocks of a Mathematical Formula

In this section, the most important commands used in mathematical typesetting will be described. Take a look at section 3.10 on page 52 for a detailed list of commands for typesetting mathematical symbols.

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

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

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

[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.

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

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

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

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.

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

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

To add mathematical accents such as small arrows or tilde signs to variables, you can use the commands given in Table 3.1 on page 52. 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}
```

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

Usually you don't typeset an explicit dot sign to indicate the multiplication operation. However sometimes it is written to help the reader's eyes in grouping a formula. Then you should use `\cdot`

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2
    {\tau}_1 \cdot {\tau}_2
\end{displaymath}
```

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$

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`
`\arcsin`	`\cosh`	`\deg`	`\gcd`	`\lg`	`\ln`	`\Pr`
`\arctan`	`\cot`	`\det`	`\hom`	`\lim`	`\log`	`\sec`
`\arg`	`\coth`	`\dim`	`\inf`	`\liminf`	`\max`	`\sin`
`\sinh`	`\sup`	`\tan`	`\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 "$a \bmod b$" 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. [3]

[3] 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 construct, e.g. you may get a similar construct to `\atop` by `\newcommand{\newatop}[2]{\genfrac{}{}{0pt}{1}{#1}{#2}}`.

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

$${n \choose k}\qquad {x \atop y+2}$$

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. [4]

```
\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 3.8 on page 54.

```
\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}
```

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

[4] $\mathcal{AMS}$-LATEX in addition has multiline super-/subscripts

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.[5]

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

$\Big((x+1) (x-1) \Big) ^{2}$

$\big(\Big(\bigg(\Bigg($ $\quad$ $\big\}\Big\}\bigg\}\Bigg\}$ $\quad$ $\big\|\Big\|\bigg\|\Bigg\|$

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 3.5.

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

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

3.4 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 spaces: `\,` for $\frac{3}{18}$ quad ($\sqcup$), `\:` for $\frac{4}{18}$ quad ($\sqcup$) and `\;` for $\frac{5}{18}$ quad ($\sqcup$). The escaped space character `\␣` generates a medium sized space and `\quad` ($\sqcup$) and `\qquad` ($\sqcup$) 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 ($\sqcup$).

```
\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}
```

$$\int\!\!\!\int_{D} g(x,y) \, \mathrm{d} x\, \mathrm{d} y$$

instead of

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

[5] 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.

Note that ‘d’ in the differential is conventionally set in roman.

$\mathcal{A}\mathcal{M}\mathcal{S}$-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 $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX) or Chapter 8 of “The LaTeX Companion” for further details.

3.5 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.

```
\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}
```

$$\begin{aligned} f(x) &= \cos x & (3.5)\\ f'(x) &= -\sin x & (3.6)\\ \int_0^x f(y)dy &= \sin x & (3.7) \end{aligned}$$

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}}
```

$$\begin{aligned} \sin x = {} & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\ & - \frac{x^7}{7!} + \cdots \qquad (3.8) \end{aligned}$$

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

$$\begin{aligned} & \cos x = 1 - \frac{x^2}{2!} + \\ & \qquad + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \qquad (3.9) \end{aligned}$$

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

It can be difficult 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).

3.6 Phantom

We can't see phantoms, but they still occupy some space in the minds of a lot of people. LaTeX is no different. We can use this for some interesting spacing tricks.

When vertically aligning text using ^ and _ LaTeX sometimes is just a little bit too helpful. Using the \phantom command you can reserve space for characters which do not show up in the final output. Best is to look at the following examples.

```
\begin{displaymath}
{}^{12}_{\phantom{1}6}\textrm{C}
\qquad \textrm{versus} \qquad
{}^{12}_{6}\textrm{C}
\end{displaymath}
```

$${}^{12}_{6}\textrm{C} \qquad \textrm{versus} \qquad {}^{12}_{6}\textrm{C}$$

```
\begin{displaymath}
\Gamma_{ij}^{\phantom{ij}k}
\qquad \textrm{versus} \qquad
\Gamma_{ij}^{k}
\end{displaymath}
```

$$\Gamma_{ij}^{k} \qquad \textrm{versus} \qquad \Gamma_{ij}^{k}$$

3.7 Math Font Size

In math mode, TeX selects the font size according to the context. Superscripts, for example, get typeset in a smaller font. If you want to typeset part of an equation in roman, don't use the \textrm command, because the font size switching mechanism will not work, as \textrm temporarily escapes to text mode. Use \mathrm instead to keep the size switching mechanism active. But pay attention, \mathrm will only work well on short items. Spaces are still not active and accented characters do not work.[6]

```
\begin{equation}
2^{\textrm{nd}} \quad
2^{\mathrm{nd}}
\end{equation}
```

$$2^{\textrm{nd}} \quad 2^{\mathrm{nd}} \tag{3.10}$$

Nevertheless, sometimes you need to tell LaTeX the correct font size. In math mode, the fontsize is set with the four commands:

[6] The AMS-LaTeX package makes the \textrm command work with size changing.

`\displaystyle` (123), `\textstyle` (123), `\scriptstyle` (123) and `\scriptscriptstyle` (123).

Changing styles also affects the way limits are displayed.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
 \frac{\displaystyle
   \sum_{i=1}^n(x_i-\overline x)
   (y_i-\overline y)}
  {\displaystyle\biggl[
 \sum_{i=1}^n(x_i-\overline x)^2
\sum_{i=1}^n(y_i-\overline y)^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\mathop{\mathrm{corr}}(X,Y)=\frac{\displaystyle\sum_{i=1}^n(x_i-\overline x)(y_i-\overline y)}{\displaystyle\biggl[\sum_{i=1}^n(x_i-\overline x)^2\sum_{i=1}^n(y_i-\overline y)^2\biggr]^{1/2}}$$

This is one of those examples in which we need larger brackets than the standard `\left[ \right]` provides.

3.8 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 3.8.1 *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.

3.9 Bold symbols

It is quite difficult to get bold symbols in LaTeX; this is probably intentional as amateur typesetters tend to overuse them. The font change command `\mathbf` gives bold letters, but these are roman (upright) whereas mathematical symbols are normally italic. There is a `\boldmath` command, but *this can only be used outside mathematics mode.* It works for symbols too.

```
\begin{displaymath}
\mu, M \qquad \mathbf{M} \qquad
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```

$$\mu, M \qquad \mathbf{M} \qquad \boldsymbol{\mu, M}$$

Notice that the comma is bold too, which may not be what is required.

The package amsbsy (included by amsmath) as well as the bm from the tools bundle make this much easier as they include a `\boldsymbol` command.

```
\begin{displaymath}
\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \qquad \boldsymbol{\mu}, \boldsymbol{M}$$

3.10 List of Mathematical Symbols

In the following tables, you find all the symbols normally accessible from *math mode*.

To use the symbols listed in Tables 3.12–3.16,[7] the package amssymb must be loaded in the preamble of the document and the AMS math fonts must be installed, on the system. If the AMS package and fonts are not installed, on your system, have a look at
`CTAN:/tex-archive/macros/latex/required/amslatex`

Table 3.1: Math Mode Accents.

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

Table 3.2: Lowercase Greek Letters.

α	`\alpha`	θ	`\theta`	o	`o`	υ	`\upsilon`
β	`\beta`	ϑ	`\vartheta`	π	`\pi`	ϕ	`\phi`
γ	`\gamma`	ι	`\iota`	ϖ	`\varpi`	φ	`\varphi`
δ	`\delta`	κ	`\kappa`	ρ	`\rho`	χ	`\chi`
ϵ	`\epsilon`	λ	`\lambda`	ϱ	`\varrho`	ψ	`\psi`
ε	`\varepsilon`	μ	`\mu`	σ	`\sigma`	ω	`\omega`
ζ	`\zeta`	ν	`\nu`	ς	`\varsigma`		
η	`\eta`	ξ	`\xi`	τ	`\tau`		

Table 3.3: Uppercase Greek Letters.

Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

[7]These tables were derived from `symbols.tex` by David Carlisle and subsequently changed extensively as suggested by Josef Tkadlec.

Table 3.4: Binary Relations.

You can produce corresponding negations by adding a `\not` command as prefix to the following symbols.

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` or `\le`	$\geq$	`\geq` or `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$	`\sqsubset` [a]	$\sqsupset$	`\sqsupset` [a]	$\Join$	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` or `\ne`

[a] Use the latexsym package to access this symbol

Table 3.5: Binary Operators.

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$\ast$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lor`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]	$\wr$	`\wr`

Table 3.6: BIG Operators.

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`	$\bigoplus$	`\bigoplus`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`			$\bigodot$	`\bigodot`
$\int$	`\int`	$\oint$	`\oint`			$\biguplus$	`\biguplus`

Table 3.7: Arrows.

$\leftarrow$	`\leftarrow` or `\gets`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\rightarrow$	`\rightarrow` or `\to`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\iff$	`\iff` (bigger spaces)	$\leadsto$	`\leadsto` [a]

[a] Use the latexsym package to access this symbol

Table 3.8: Delimiters.

<table>
<tr><td>$($</td><td><code>(</code></td><td>$)$</td><td><code>)</code></td><td>$\uparrow$</td><td><code>\uparrow</code></td><td>$\Uparrow$</td><td><code>\Uparrow</code></td></tr>
<tr><td>$[$</td><td><code>[</code> or <code>\lbrack</code></td><td>$]$</td><td><code>]</code> or <code>\rbrack</code></td><td>$\downarrow$</td><td><code>\downarrow</code></td><td>$\Downarrow$</td><td><code>\Downarrow</code></td></tr>
<tr><td>$\{$</td><td><code>\{</code> or <code>\lbrace</code></td><td>$\}$</td><td><code>\}</code> or <code>\rbrace</code></td><td>$\updownarrow$</td><td><code>\updownarrow</code></td><td>$\Updownarrow$</td><td><code>\Updownarrow</code></td></tr>
<tr><td>$\langle$</td><td><code>\langle</code></td><td>$\rangle$</td><td><code>\rangle</code></td><td>$\vert$</td><td><code>|</code> or <code>\vert</code></td><td>$\Vert$</td><td><code>\|</code> or <code>\Vert</code></td></tr>
<tr><td>$\lfloor$</td><td><code>\lfloor</code></td><td>$\rfloor$</td><td><code>\rfloor</code></td><td>$\lceil$</td><td><code>\lceil</code></td><td>$\rceil$</td><td><code>\rceil</code></td></tr>
<tr><td>$/$</td><td><code>/</code></td><td>$\backslash$</td><td><code>\backslash</code></td><td></td><td>. (dual. empty)</td><td></td><td></td></tr>
</table>

Table 3.9: Large Delimiters.

$\lgroup$	`\lgroup`	$\rgroup$	`\rgroup`	$\lmoustache$	`\lmoustache`	$\rmoustache$	`\rmoustache`
$\arrowvert$	`\arrowvert`	$\Arrowvert$	`\Arrowvert`	$\bracevert$	`\bracevert`		

Table 3.10: Miscellaneous Symbols.

$\dots$	\dots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph	$\wp$	\wp
$\forall$	\forall	$\exists$	\exists	$\mho$	\mho [a]	∂	\partial
$'$	'	$\prime$	\prime	$\emptyset$	\emptyset	∞	\infty
∇	\nabla	$\triangle$	\triangle	$\Box$	\Box [a]	$\Diamond$	\Diamond [a]
$\bot$	\bot	$\top$	\top	$\angle$	\angle	$\surd$	\surd
$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit	$\clubsuit$	\clubsuit	$\spadesuit$	\spadesuit
$\neg$	\neg or \lnot	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp

[a] Use the latexsym package to access this symbol

Table 3.11: Non-Mathematical Symbols.

These symbols can also be used in text mode.

†	\dag	§	\S	©	\copyright
‡	\ddag	¶	\P	£	\pounds

Table 3.12: AMS Delimiters.

$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner	$\lrcorner$	\lrcorner
$\lvert$	\lvert	$\rvert$	\rvert	$\lVert$	\lVert	$\rVert$	\rVert

Table 3.13: AMS Greek and Hebrew.

$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth	$\daleth$	\daleth	$\gimel$	\gimel

Table 3.14: AMS Binary Relations.

$\lessdot$	`\lessdot`	$\gtrdot$	`\gtrdot`	$\doteqdot$	`\doteqdot` or `\Doteq`
$\leqslant$	`\leqslant`	$\geqslant$	`\geqslant`	$\risingdotseq$	`\risingdotseq`
$\eqslantless$	`\eqslantless`	$\eqslantgtr$	`\eqslantgtr`	$\fallingdotseq$	`\fallingdotseq`
$\leqq$	`\leqq`	$\geqq$	`\geqq`	$\eqcirc$	`\eqcirc`
$\lll$	`\lll` or `\llless`	$\ggg$	`\ggg` or `\gggtr`	$\circeq$	`\circeq`
$\lesssim$	`\lesssim`	$\gtrsim$	`\gtrsim`	$\triangleq$	`\triangleq`
$\lessapprox$	`\lessapprox`	$\gtrapprox$	`\gtrapprox`	$\bumpeq$	`\bumpeq`
$\lessgtr$	`\lessgtr`	$\gtrless$	`\gtrless`	$\Bumpeq$	`\Bumpeq`
$\lesseqgtr$	`\lesseqgtr`	$\gtreqless$	`\gtreqless`	$\thicksim$	`\thicksim`
$\lesseqqgtr$	`\lesseqqgtr`	$\gtreqqless$	`\gtreqqless`	$\thickapprox$	`\thickapprox`
$\preccurlyeq$	`\preccurlyeq`	$\succcurlyeq$	`\succcurlyeq`	$\approxeq$	`\approxeq`
$\curlyeqprec$	`\curlyeqprec`	$\curlyeqsucc$	`\curlyeqsucc`	$\backsim$	`\backsim`
$\precsim$	`\precsim`	$\succsim$	`\succsim`	$\backsimeq$	`\backsimeq`
$\precapprox$	`\precapprox`	$\succapprox$	`\succapprox`	$\vDash$	`\vDash`
$\subseteqq$	`\subseteqq`	$\supseteqq$	`\supseteqq`	$\Vdash$	`\Vdash`
$\Subset$	`\Subset`	$\Supset$	`\Supset`	$\Vvdash$	`\Vvdash`
$\sqsubset$	`\sqsubset`	$\sqsupset$	`\sqsupset`	$\backepsilon$	`\backepsilon`
$\therefore$	`\therefore`	$\because$	`\because`	$\varpropto$	`\varpropto`
$\shortmid$	`\shortmid`	$\shortparallel$	`\shortparallel`	$\between$	`\between`
$\smallsmile$	`\smallsmile`	$\smallfrown$	`\smallfrown`	$\pitchfork$	`\pitchfork`
$\vartriangleleft$	`\vartriangleleft`	$\vartriangleright$	`\vartriangleright`	$\blacktriangleleft$	`\blacktriangleleft`
$\trianglelefteq$	`\trianglelefteq`	$\trianglerighteq$	`\trianglerighteq`	$\blacktriangleright$	`\blacktriangleright`

Table 3.15: AMS Arrows.

$\dashleftarrow$	`\dashleftarrow`	$\dashrightarrow$	`\dashrightarrow`	$\multimap$	`\multimap`
$\leftleftarrows$	`\leftleftarrows`	$\rightrightarrows$	`\rightrightarrows`	$\upuparrows$	`\upuparrows`
$\leftrightarrows$	`\leftrightarrows`	$\rightleftarrows$	`\rightleftarrows`	$\downdownarrows$	`\downdownarrows`
$\Lleftarrow$	`\Lleftarrow`	$\Rrightarrow$	`\Rrightarrow`	$\upharpoonleft$	`\upharpoonleft`
$\twoheadleftarrow$	`\twoheadleftarrow`	$\twoheadrightarrow$	`\twoheadrightarrow`	$\upharpoonright$	`\upharpoonright`
$\leftarrowtail$	`\leftarrowtail`	$\rightarrowtail$	`\rightarrowtail`	$\downharpoonleft$	`\downharpoonleft`
$\leftrightharpoons$	`\leftrightharpoons`	$\rightleftharpoons$	`\rightleftharpoons`	$\downharpoonright$	`\downharpoonright`
$\Lsh$	`\Lsh`	$\Rsh$	`\Rsh`	$\rightsquigarrow$	`\rightsquigarrow`
$\looparrowleft$	`\looparrowleft`	$\looparrowright$	`\looparrowright`	$\leftrightsquigarrow$	`\leftrightsquigarrow`
$\curvearrowleft$	`\curvearrowleft`	$\curvearrowright$	`\curvearrowright`		
$\circlearrowleft$	`\circlearrowleft`	$\circlearrowright$	`\circlearrowright`		

Table 3.16: AMS Negated Binary Relations and Arrows.

$\nless$	\nless	$\ngtr$	\ngtr	$\varsubsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\varsupsetneqq$	\varsupsetneqq
$\nleq$	\nleq	$\ngeq$	\ngeq	$\nsubseteqq$	\nsubseteqq
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant	$\nsupseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\nsim$	\nsim
$\nprec$	\nprec	$\nsucc$	\nsucc	$\ncong$	\ncong
$\npreceq$	\npreceq	$\nsucceq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\ntriangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\ntriangleright$	\ntriangleright
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\ntrianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\ntrianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

Table 3.17: AMS Binary Operators.

$\dotplus$	\dotplus	$\centerdot$	\centerdot	$\intercal$	\intercal
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\Cup$	\Cup or \doublecup	$\Cap$	\Cap or \doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes	$\circledast$	\circledast
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge		

Table 3.18: AMS Miscellaneous.

$\hbar$	`\hbar`	$\hslash$	`\hslash`	$\Bbbk$	`\Bbbk`
$\square$	`\square`	$\blacksquare$	`\blacksquare`	$\circledS$	`\circledS`
$\vartriangle$	`\vartriangle`	$\blacktriangle$	`\blacktriangle`	$\complement$	`\complement`
$\triangledown$	`\triangledown`	$\blacktriangledown$	`\blacktriangledown`	$\Game$	`\Game`
$\lozenge$	`\lozenge`	$\blacklozenge$	`\blacklozenge`	$\bigstar$	`\bigstar`
$\angle$	`\angle`	$\measuredangle$	`\measuredangle`	$\sphericalangle$	`\sphericalangle`
$\diagup$	`\diagup`	$\diagdown$	`\diagdown`	$\backprime$	`\backprime`
$\nexists$	`\nexists`	$\Finv$	`\Finv`	$\varnothing$	`\varnothing`
$\eth$	`\eth`	$\mho$	`\mho`		

Table 3.19: Math Alphabets.

Example	Command	Required package
ABCdef	`\mathrm{ABCdef}`	
ABCdef	`\mathit{ABCdef}`	
$\mathnormal{ABCdef}$	`\mathnormal{ABCdef}`	
$\mathcal{ABC}$	`\mathcal{ABC}`	
$\mathscr{ABC}$	`\mathcal{ABC}`	mathrsfs
$\mathcal{ABC}$	`\mathcal{ABC}`	eucal with option: mathcal or
	`\mathscr{ABC}`	eucal with option: mathscr
$\mathfrak{ABCdef}$	`\mathfrak{ABCdef}`	eufrak
$\mathbb{ABC}$	`\mathbb{ABC}`	amsfonts or amssymb

Chapter 4

Specialities

When putting together a large document, LaTeX will help you with some special features like index generation, bibliography management, and other things. A much more complete description of specialities and enhancements possible with LaTeX can be found in the *LaTeX Manual* [1] and *The LaTeX Companion* [3].

4.1 Including EPS Graphics

LaTeX provides the basic facilities to work with floating bodies such as images or graphics, with the `figure` and the `table` environment.

There are also several possibilities to generate the actual graphics with basic LaTeX or a LaTeX extension package. Unfortunately, most users find them quite difficult to understand. Therefore this will not be explained any further in this manual. Please refer to *The LaTeX Companion* [3] and the *LaTeX Manual* [1] for more information on that subject.

A much easier way to get graphics into a document, is to generate them with a specialised software package[1] and then include the finished graphics into the document. Here again, LaTeX packages offer many ways to do that. In this introduction, only the use of Encapsulated PostScript (EPS) graphics will be discussed, because it is quite easy to do and widely used. In order to use pictures in the EPS format, you must have a PostScript printer[2] available for output.

A good set of commands for inclusion of graphics is provided in the graphicx package by D. P. Carlisle. It is part of a whole family of packages called the "graphics" bundle[3].

[1] Such as XFig, CorelDraw!, Freehand, Gnuplot, ...

[2] Another possibility to output PostScript is the GhostScript program available from `CTAN:/tex-archive/support/ghostscript`. Windows and OS/2 users might want to look for GSView.

[3] `CTAN:/tex-archive/macros/latex/required/graphics`

Assuming you are working on a system with a PostScript printer available for output and with the graphicx package installed, you can use the following step by step guide to include a picture into your document:

1. Export the picture from your graphics program in EPS format.[4]

2. Load the graphicx package in the preamble of the input file with

   `\usepackage[`*driver*`]{graphicx}`

   where *driver* is the name of your "dvi to postscript" converter program. The most widely used program is called `dvips`. The name of the driver is required, because there is no standard on how graphics are included in TeX. Knowing the name of the *driver*, the graphicx package can choose the correct method to insert information about the graphics into the `.dvi` file, so that the printer understands it and can correctly include the `.eps` file.

3. Use the command

   `\includegraphics[`*key*=*value*, ...`]{`*file*`}`

   to include *file* into your document. The optional parameter accepts a comma separated list of *keys* and associated *values*. The *keys* can be used to alter the width, height and rotation of the included graphic. Table 4.1 lists the most important keys.

Table 4.1: Key Names for graphicx Package.

`width`	scale graphic to the specified width
`height`	scale graphic to the specified height
`angle`	rotate graphic counterclockwise
`scale`	scale graphic

[4] If your software can not export into EPS format, you can try to install a PostScript printer driver (some Apple LaserWriter for example) and then print to a file with this driver. With some luck this file will be in EPS format. Note that an EPS must not contain more than one page. Some printer drivers can be explicitly configured to produce EPS format.

The following example code will hopefully make things clear:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test}
\end{center}
\end{figure}
```

It includes the graphic stored in the file `test.eps`. The graphic is *first* rotated by an angle of 90 degrees and *then* scaled to the final width of 0.5 times the width of a standard paragraph. The aspect ratio is 1.0, because no special height is specified. The width and height parameters can also be specified in absolute dimensions. Refer to Table 5.5 on page 76 for more information. If you want to know more about this topic, make sure to read [8] and [11].

4.2 Bibliography

You can produce a bibliography with the `thebibliography` environment. Each entry starts with

`\bibitem{`*marker*`}`

The *marker* is then used to cite the book, article or paper within the document.

`\cite{`*marker*`}`

The numbering of the entries is generated automatically. The parameter after the `\begin{thebibliography}` command sets the maximum width of these numbers. In the example below, `{99}` tells LaTeX to expect that none of the bibliography item numbers will be wider than the number 99.

```
Partl~\cite{pa} has
proposed that \ldots
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Volume~9, Issue~1 (1988)
\end{thebibliography}
```

Partl [1] has proposed that . . .

Bibliography

[1] H. Partl: *German TeX*, TUGboat Volume 9, Issue 1 (1988)

For larger projects, you might want to check out the BibTEX program. BibTEX is included with most TEX distributions. It allows you to maintain a bibliographic database and then extract the references relevant to things you cited in your paper. The visual presentation of BibTEX generated bibliographies is based on a style sheets concept which allows you to create bibliographies following a wide range of established designs.

4.3 Indexing

A very useful feature of many books is their index. With LaTeX and the support program `makeindex`[5], an index can be generated quite easily. In this introduction, only the basic index generation commands will be explained. For a more in-depth view, please refer to *The LaTeX Companion* [3].

To enable the indexing feature of LaTeX, the makeidx package must be loaded in the preamble with:

```
\usepackage{makeidx}
```

and the special indexing commands must be enabled by putting the

```
\makeindex
```

command into the input file preamble.

The content of the index is specified with

```
\index{key}
```

commands, where *key* is the index entry. You enter the index commands at the points in the text where you want the final index entries to point to. Table 4.2 explains the syntax of the *key* argument with several examples.

When the input file is processed with LaTeX, each `\index` command writes an appropriate index entry together with the current page number to a special file. The file has the same name as the LaTeX input file, but a different extension (`.idx`). This `.idx` file can then be processed with the `makeindex` program.

```
makeindex filename
```

The `makeindex` program generates a sorted index with the same base file name, but this time with the extension `.ind`. If now the LaTeX input

[5] On systems not necessarily supporting filenames longer than 8 characters, the program may be called `makeidx`.

Table 4.2: Index Key Syntax Examples.

Example	**Index Entry**	**Comment**
`\index{hello}`	hello, 1	Plain entry
`\index{hello!Peter}`	Peter, 3	Subentry under 'hello'
`\index{Sam@\textsl{Sam}}`	*Sam*, 2	Formatted entry
`\index{Lin@\textbf{Lin}}`	**Lin**, 7	Same as above
`\index{Jenny|textbf}`	Jenny, **3**	Formatted page number
`\index{Joe|textit}`	Joe, *5*	Same as above

file is processed again, this sorted index gets included into the document at the point where LaTeX finds

```
\printindex
```

The showidx package which comes with LaTeX 2ε prints out all index entries in the left margin of the text. This is quite useful for proofreading a document and verifying the index.

4.4 Fancy Headers

The fancyhdr package,[6] written by Piet van Oostrum, provides a few simple commands which allow you to customize the header and footer lines of your document. If you look at the top of this page, you can see a possible application of this package.

The tricky problem when customising headers and footers is to get things like running section and chapter names in there. LaTeX accomplishes this with a two-stage approach. In the header and footer definition, you use the commands `\rightmark` and `\leftmark` to represent the current section and chapter heading, respectively. The values of these two commands are overwritten whenever a chapter or section command is processed.

For ultimate flexibility, the `\chapter` command and its friends do not redefine `\rightmark` and `\leftmark` themselves, they call yet another command called `\chaptermark`, `\sectionmark` or `\subsectionmark` which is responsible for redefining `\rightmark` and `\leftmark`.

So, if you wanted to change the look of the chapter name in the header line, you simply have to "renew" the `\chaptermark` command.

Figure 4.1 shows a possible setup for the fancyhdr package which makes the headers look about the same as they look in this booklet. In any case

[6]Available from `CTAN:/tex-archive/macros/latex/contrib/supported/fancyhdr`.

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% with this we ensure that the chapter and section
% headings are in lowercase.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}  % delete current setting for header and footer
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % make space for the rule
\fancypagestyle{plain}{%
   \fancyhead{} % get rid of headers on plain pages
   \renewcommand{\headrulewidth}{0pt} % and the line
}
```

Figure 4.1: Example fancyhdr Setup.

I suggest you fetch the documentation for the package at the address mentioned in the footnote.

4.5 The Verbatim Package

Earlier in this book, you got to know the `verbatim` *environment*. In this section, you are going to learn about the verbatim *package*. The verbatim package is basically a re-implementation of the `verbatim` environment, which works around some of the limitations of the original `verbatim` environment. This by itself is not spectacular, but with the implementation of the verbatim package, there was also new functionality added, and this is the reason I am mentioning the package here. The verbatim package provides the

`\verbatiminput{`*filename*`}`

command which allows you to include raw ASCII text into your document as if it was inside a `verbatim` environment.

As the verbatim package is part of the 'tools' bundle, you should find it preinstalled on most systems. If you want to know more about this package, make sure to read [9]

4.6 Downloading and Installing LaTeX Packages

Most LaTeX installations come with a large set of pre-installed style packages, but there are many more available on the net. The main place to look for style package on the Internet is CTAN (`http://www.ctan.org/`).

Packages, such as, geometry, hyphenat, and many others, are typically made up of two files: a file with the extension `.ins` and another with the extension `.dtx`. Often there will be a `readme.txt` with a brief description of the package. You should of course read this file first.

In any event, once you have copied the package files onto your machine, you still have to process them in a way that (a) your TeX distribution knows about the new style package and (b) you get the documentation. Here's how you do the first part:

1. Run LaTeX on the `.ins` file. This will extract a `.sty` file.
2. Move the `.sty` file to a place where your distribution can find it. Usually this is in your `.../`*`localtexmf`*`/tex/latex` subdirectory (Windows or OS/2 users should feel free to change the direction of the slashes).
3. Refresh your distribution's file-name database. The command depends on the LaTeX-Distribution you use: teTeX, fpTeX – `texhash`; web2c – `maktexlsr`; MikTeX – `initexmf -update-fndb` or use the GUI.

Now you can extract the documentation from the `.dtx` file:

1. Run LaTeX on the `.dtx` file. This will generate a `.dvi` file. Note that you may have to run LaTeX several times before it gets the cross-references right.
2. Check to see if LaTeX has produced a `.idx` file among the various files you now have. If you do not see this file, then you may proceed to step 5.
3. In order to generate the index, type the following:
   `makeindex -s gind.ist` *`name`*
   (where *name* stands for the main-file name without any extension).
4. Run LaTeX on the `.dtx` file once again.
5. Last but not least, make a `.ps` or `.pdf` file to increase your reading pleasure.

Sometimes you will see that a `.glo` (glossary) file has been produced. Run the following command between step 4 and 5:
`makeindex -s gglo.ist -o` *`name`*`.gls` *`name`*`.glo`
Be sure to run LaTeX on the `.dtx` one last time before moving on to step 5.

Chapter 5

Customising LaTeX

Documents produced by using the commands you have learned up to this point will look acceptable to a large audience. While they are not looking fancy, they obey all the established rules of good typesetting, which will make them easy to read and pleasant to look at.

However there are situations where LaTeX does not provide a command or environment which matches your needs, or the output produced by some existing command may not meet your requirements.

In this chapter, I will try to give some hints on how to teach LaTeX new tricks and how to make it produce output which looks different than what is provided by default.

5.1 New Commands, Environments and Packages

You may have noticed that all the commands I introduce in this book are typeset in a box, and that they show up in the index at the end of the book. Instead of directly using the necessary LaTeX commands to achieve this, I have created a package in which I defined new commands and environments for this purpose. Now I can simply write:

```
\begin{lscommand}
\ci{dum}
\end{lscommand}
```

`\dum`

In this example, I am using both a new environment called `lscommand` which is responsible for drawing the box around the command and a new command named `\ci` which typesets the command name and also makes a corresponding entry in the index. You can check this out by looking up the `\dum` command in the index at the back of this book, where you'll find an entry for `\dum`, pointing to every page where I mentioned the `\dum` command.

If I ever decide that I do not like the commands to be typeset in a box any more, I can simply change the definition of the `lscommand` environment to create a new look. This is much easier than going through the whole document to hunt down all the places where I have used some generic LaTeX commands to draw a box around some word.

5.1.1 New Commands

To add your own commands, use the

`\newcommand{`*name*`}[`*num*`]{`*definition*`}`

command. Basically, the command requires two arguments: the *name* of the command you want to create, and the *definition* of the command. The *num* argument in square brackets is optional and specifies the number of arguments the new command takes (up to 9 are possible). If missing it defaults to 0, i.e. no argument allowed.

The following two examples should help you to get the idea. The first example defines a new command called `\tnss`. This is short for "The Not So Short Introduction to LaTeX 2_ε". Such a command could come in handy if you had to write the title of this book over and over again.

```
\newcommand{\tnss}{The not
    so Short Introduction to
    \LaTeXe}
This is ``\tnss'' \ldots{}
``\tnss''
```

This is "The not so Short Introduction to LaTeX 2_ε" ... "The not so Short Introduction to LaTeX 2_ε"

The next example illustrates how to define a new command which takes one argument. The `#1` tag gets replaced by the argument you specify. If you wanted to use more than one argument, use `#2` and so on.

```
\newcommand{\txsit}[1]
 {This is the \emph{#1} Short
      Introduction to \LaTeXe}
% in the document body:
\begin{itemize}
\item \txsit{not so}
\item \txsit{very}
\end{itemize}
```

- This is the *not so* Short Introduction to LaTeX 2_ε
- This is the *very* Short Introduction to LaTeX 2_ε

LaTeX will not allow you to create a new command which would overwrite an existing one. But there is a special command in case you explicitly want this: `\renewcommand`. It uses the same syntax as the `\newcommand` command.

In certain cases you might also want to use the `\providecommand` command. It works like `\newcommand`, but if the command is already defined, LaTeX 2_ε will silently ignore it.

There are some points to note about whitespace following LaTeX commands. See page 6 for more information.

5.1.2 New Environments

Similar to the `\newcommand` command, there is also a command to create your own environments. The `\newenvironment` command uses the following syntax:

`\newenvironment{`*name*`}[`*num*`]{`*before*`}{`*after*`}`

Like the `\newcommand` command, you can use `\newenvironment` with an optional argument or without. The material specified in the *before* argument is processed before the text in the environment gets processed. The material in the *after* argument gets processed when the `\end{`*name*`}` command is encountered.

The example below illustrates the usage of the `\newenvironment` command.

```
\newenvironment{king}
 {\rule{1ex}{1ex}%
      \hspace{\stretch{1}}}
 {\hspace{\stretch{1}}%
      \rule{1ex}{1ex}}

\begin{king}
My humble subjects \ldots
\end{king}
```

■ My humble subjects … ■

The *num* argument is used the same way as in the `\newcommand` command. LaTeX makes sure that you do not define an environment which already exists. If you ever want to change an existing command, you can use the `\renewenvironment` command. It uses the same syntax as the `\newenvironment` command.

The commands used in this example will be explained later: For the `\rule` command see page 81, for `\stretch` go to page 75, and more information on `\hspace` can be found on page 75.

5.1.3 Your own Package

If you define a lot of new environments and commands, the preamble of your document will get quite long. In this situation, it is a good idea to create a LaTeX package containing all your command and environment definitions. You can then use the `\usepackage` command to make the package available in your document.

```
% Demo Package by Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\tnss}{The not so Short Introduction to \LaTeXe}
\newcommand{\txsit}[1]{The \emph{#1} Short
                        Introduction to \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

Figure 5.1: Example Package.

Writing a package consists basically in copying the contents of your document preamble into a separate file with a name ending in `.sty`. There is one special command,

`\ProvidesPackage{`*package name*`}`

for use at the very beginning of your package file. `\ProvidesPackage` tells LaTeX the name of the package and will allow it to issue a sensible error message when you try to include a package twice. Figure 5.1 shows a small example package which contains the commands defined in the examples above.

5.2 Fonts and Sizes

5.2.1 Font changing Commands

LaTeX chooses the appropriate font and font size based on the logical structure of the document (sections, footnotes, ...). In some cases, one might like to change fonts and sizes by hand. To do this, you can use the commands listed in Tables 5.1 and 5.2. The actual size of each font is a design issue and depends on the document class and its options. Table 5.3 shows the absolute point size for these commands as implemented in the standard document classes.

```
{\small The small and
\textbf{bold} Romans ruled}
{\Large all of great big
\textit{Italy}.}
```

The small and **bold** Romans ruled all of great big *Italy*.

One important feature of LaTeX 2_ε is, that the font attributes are independent. This means, that you can issue size or even font changing commands and still keep the bold or slant attribute set earlier.

In *math mode* you can use the font changing *commands* to temporarily exit *math mode* and enter some normal text. If you want to switch to another font for math typesetting there exists another special set of commands. Refer to Table 5.4.

In connection with the font size commands, curly braces play a significant role. They are used to build *groups*. Groups limit the scope of most LaTeX commands.

```
He likes {\LARGE large and
{\small small} letters}.
```

He likes large and small letters.

The font size commands also change the line spacing, but only if the paragraph ends within the scope of the font size command. The closing curly brace `}` should therefore not come too early. Note the position of the `\par` command in the next two examples. [1]

[1] `\par` is equivalent to a blank line

Table 5.1: Fonts.

`\textrm{...}`	roman	`\textsf{...}`	sans serif
`\texttt{...}`	`typewriter`		
`\textmd{...}`	medium	`\textbf{...}`	**bold face**
`\textup{...}`	upright	`\textit{...}`	*italic*
`\textsl{...}`	*slanted*	`\textsc{...}`	SMALL CAPS
`\emph{...}`	*emphasized*	`\textnormal{...}`	document font

Table 5.2: Font Sizes.

`\tiny`	tiny font	`\Large`	larger font
`\scriptsize`	very small font	`\LARGE`	very large font
`\footnotesize`	quite small font	`\huge`	huge
`\small`	small font	`\Huge`	largest
`\normalsize`	normal font		
`\large`	large font		

Table 5.3: Absolute Point Sizes in Standard Classes.

size	10pt (default)	11pt option	12pt option
`\tiny`	5pt	6pt	6pt
`\scriptsize`	7pt	8pt	8pt
`\footnotesize`	8pt	9pt	10pt
`\small`	9pt	10pt	11pt
`\normalsize`	10pt	11pt	12pt
`\large`	12pt	12pt	14pt
`\Large`	14pt	14pt	17pt
`\LARGE`	17pt	17pt	20pt
`\huge`	20pt	20pt	25pt
`\Huge`	25pt	25pt	25pt

Table 5.4: Math Fonts.

Command	*Example*	*Output*
`\mathcal{...}`	`$\mathcal{B}=c$`	$\mathcal{B}=c$
`\mathrm{...}`	`$\mathrm{K}_2$`	K_2
`\mathbf{...}`	`$\sum x=\mathbf{v}$`	$\sum x=\mathbf{v}$
`\mathsf{...}`	`$\mathsf{G\times R}$`	$\mathsf{G\times R}$
`\mathtt{...}`	`$\mathtt{L}(b,c)$`	$\mathtt{L}(b,c)$
`\mathnormal{...}`	`$\mathnormal{R_{19}}\neq R_{19}$`	$\mathnormal{R_{19}}\neq R_{19}$
`\mathit{...}`	`$\mathit{ffi}\neq ffi$`	$\mathit{ffi}\neq ffi$

```
{\Large Don't read this! It is not
true. You can believe me!\par}
```

Don't read this! It is not true. You can believe me!

```
{\Large This is not true either.
But remember I am a liar.}\par
```

This is not true either. But remember I am a liar.

If you want to activate a size changing command for a whole paragraph of text or even more, you might want to use the environment syntax for font changing commands.

```
\begin{Large}
This is not true.
But then again, what is these
days \ldots
\end{Large}
```

This is not true. But then again, what is these days ...

This will save you from counting lots of curly braces.

5.2.2 Danger, Will Robinson, Danger

As noted at the beginning of this chapter, it is dangerous to clutter your document with explicit commands like this, because they work in opposition to the basic idea of LaTeX, which is to separate the logical and visual markup of your document. This means that if you use the same font changing command in several places in order to typeset a special kind of information, you should use `\newcommand` to define a "logical wrapper command" for the font changing command.

```
\newcommand{\oops}[1]{\textbf{#1}}
Do not \oops{enter} this room,
it's occupied by a \oops{machine}
of unknown origin and purpose.
```

Do not **enter** this room, it's occupied by a **machine** of unknown origin and purpose.

This approach has the advantage that you can decide at some later stage whether you want to use some other visual representation of danger than `\textbf` without having to wade through your document, identifying all the occurrences of `\textbf` and then figuring out for each one whether it was used for pointing out danger or for some other reason.

5.2.3 Advice

To conclude this journey into the land of fonts and font sizes, here is a little word of advice:

> **Remember!** *The* MO R*E* fonts you use **in** a `document`, *the* more READABLE and *beautiful it* becomeS.

5.3 Spacing

5.3.1 Line Spacing

If you want to use larger inter-line spacing in a document, you can change its value by putting the

`\linespread{`*factor*`}`

command into the preamble of your document. Use `\linespread{1.3}` for "one and a half" line spacing, and `\linespread{1.6}` for "double" line spacing. Normally the lines are not spread, therefore the default line spread factor is 1.

5.3.2 Paragraph Formatting

In LaTeX, there are two parameters influencing paragraph layout. By placing a definition like

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

in the preamble of the input file, you can change the layout of paragraphs. These two commands increase the space between two paragraphs while setting the paragraph indent to zero.

The `plus` and `minus` parts of the length above tell TeX, that it can compress and expand the inter paragraph skip by the amount specified if this is necessary to properly fit the paragraphs onto the page.

In continental Europe, paragraphs are often separated by some space and not indented. But beware, this also has its effect on the table of contents. Its lines get spaced more loosely now as well. To avoid this, you might want to move the two commands from the preamble into your document to some place after the `\tableofcontents` or to not use them at all, because you'll find that most professional books use indenting and not spacing to separate paragraphs.

If you want to indent a paragraph which is not indented, you can use

```
\indent
```

at the beginning of the paragraph.[2] Obviously, this will only have an effect when `\parindent` is not set to zero.

To create a non-indented paragraph, you can use

```
\noindent
```

as the first command of the paragraph. This might come in handy when you start a document with body text and not with a sectioning command.

5.3.3 Horizontal Space

LaTeX determines the spaces between words and sentences automatically. To add horizontal space, use:

```
\hspace{length}
```

If such a space should be kept even if it falls at the end or the start of a line, use `\hspace*` instead of `\hspace`. The *length* in the simplest case just is a number plus a unit. The most important units are listed in Table 5.5.

```
This\hspace{1.5cm}is a space
of 1.5 cm.
```

This   is a space of 1.5 cm.

The command

```
\stretch{n}
```

generates a special rubber space. It stretches until all the remaining space on a line is filled up. If two `\hspace{\stretch{n}}` commands are issued on the same line, they grow according to the stretch factor.

```
x\hspace{\stretch{1}}
x\hspace{\stretch{3}}x
```

x   x   x

[2] To indent the first paragraph after each section head, use the indentfirst package in the 'tools' bundle.

Table 5.5: TeX Units.

`mm`	millimetre $\approx$ 1/25 inch
`cm`	centimetre = 10 mm
`in`	inch = 25.4 mm
`pt`	point $\approx$ 1/72 inch $\approx \frac{1}{3}$ mm
`em`	approx width of an 'M' in the current font
`ex`	approx height of an 'x' in the current font

5.3.4 Vertical Space

The space between paragraphs, sections, subsections, ... is determined automatically by LaTeX. If necessary, additional vertical space *between two paragraphs* can be added with the command:

`\vspace{`*length*`}`

This command should normally be used between two empty lines. If the space should be preserved at the top or at the bottom of a page, use the starred version of the command `\vspace*` instead of `\vspace`.

The `\stretch` command in connection with `\pagebreak` can be used to typeset text on the last line of a page, or to centre text vertically on a page.

```
Some text \ldots

\vspace{\stretch{1}}
This goes onto the last line of the page.\pagebreak
```

Additional space between two lines of *the same* paragraph or within a table is specified with the

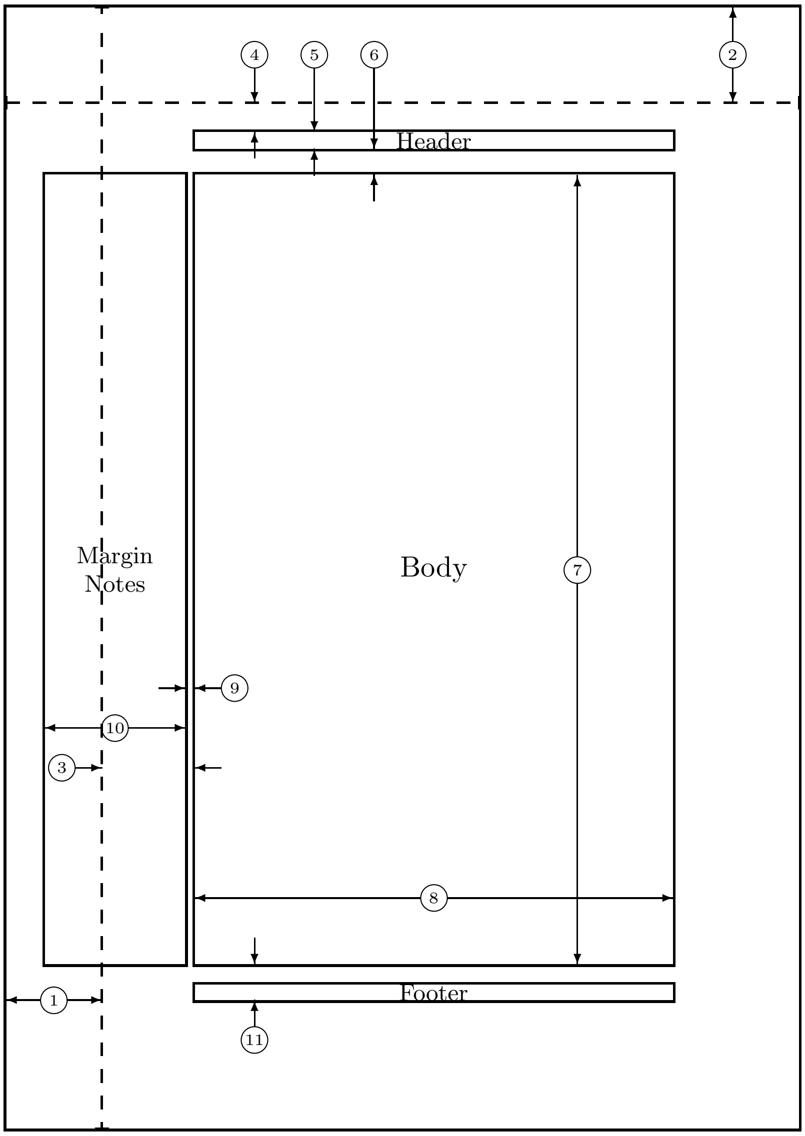
`\\[`*length*`]`

command.

With `\bigskip` and `\smallskip` you can skip a predefined amount of vertical space without having to worry about exact numbers.

5.4 Page Layout

LaTeX 2_ε allows you to specify the paper size in the `\documentclass` command. It then automatically picks the right text margins. But sometimes you may not be happy with the predefined values. Naturally, you can change



1	`one inch + \hoffset`	2	`one inch + \voffset`
3	`\evensidemargin = 70pt`	4	`\topmargin = 22pt`
5	`\headheight = 13pt`	6	`\headsep = 19pt`
7	`\textheight = 595pt`	8	`\textwidth = 360pt`
9	`\marginparsep = 7pt`	10	`\marginparwidth = 106pt`
11	`\footskip = 27pt`		`\marginparpush = 5pt (not shown)`
	`\hoffset = 0pt`		`\voffset = 0pt`
	`\paperwidth = 597pt`		`\paperheight = 845pt`

Figure 5.2: Page Layout Parameters.

them. Figure 5.2 shows all the parameters which can be changed. The figure was produced with the layout package from the tools bundle[3].

WAIT! ... before you launch into a "Let's make that narrow page a bit wider" frenzy, take a few seconds to think. As with most things in LaTeX, there is a good reason for the page layout to be as it is.

Sure, compared to your off-the-shelf MS Word page, it looks awfully narrow. But take a look at your favourite book[4] and count the number of characters on a standard text line. You will find that there are no more than about 66 characters on each line. Now do the same on your LaTeX page. You will find that there are also about 66 characters per line. Experience shows that the reading gets difficult as soon as there are more characters on a single line. This is because it is difficult for the eyes to move from the end of one line to the start of the next one. This is also the reason why newspapers are typeset in multiple columns.

So if you increase the width of your body text, keep in mind that you are making life difficult for the readers of your paper. But enough of the cautioning, I promised to tell you how you do it ...

LaTeX provides two commands to change these parameters. They are usually used in the document preamble.

The first command assigns a fixed value to any of the parameters:

`\setlength{`*parameter*`}{`*length*`}`

The second command adds a length to any of the parameters.

`\addtolength{`*parameter*`}{`*length*`}`

This second command is actually more useful than the `\setlength` command, because you can now work relative to the existing settings. To add one centimetre to the overall text width, I put the following commands into the document preamble:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

In this context, you might want to look at the calc package, it allows you to use arithmetic operations in the argument of setlength and other places where you can enter numeric values into function arguments.

5.5 More fun with lengths

Whenever possible, I avoid using absolute lengths in LaTeX documents. I rather try to base things on the width or height of other page elements. For

[3] `CTAN:/tex-archive/macros/latex/required/tools`

[4] I mean a real printed book produced by a reputable publisher.

the width of a figure this could be `\textwidth` in order to make it fill the page.

The following 3 commands allow you to determine the width, height and depth of a text string.

```
\settoheight{lscommand}{text}
\settodepth{lscommand}{text}
\settowidth{lscommand}{text}
```

The example below shows a possible application of these commands.

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[0pt][r]{#1:\ }}{}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Where}$a$,
$b$ -- are adjunct to the right
angle of a right-angled triangle.

$c$ -- is the hypotenuse of
the triangle and feels lonely.

$d$ -- finally does not show up
here at all. Isn't that puzzling?
\end{vardesc}
```

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

Where: a, b – are adjunct to the right angle of a right-angled triangle.

c – is the hypotenuse of the triangle and feels lonely.

d – finally does not show up here at all. Isn't that puzzling?

5.6 Boxes

LaTeX builds up its pages by pushing around boxes. At first, each letter is a little box, which is then glued to other letters to form words. These are again glued to other words, but with special glue, which is elastic so that a series of words can be squeezed or stretched as to exactly fill a line on the page.

I admit, this is a very simplistic version of what really happens, but the point is that TeX operates on glue and boxes. Not only a letter can be a box. You can put virtually everything into a box including other boxes. Each box will then be handled by LaTeX as if it was a single letter.

In the past chapters you have already encountered some boxes, although I did not tell you. The `tabular` environment and the `\includegraphics`, for example, both produce a box. This means that you can easily arrange two tables or images side by side. You just have to make sure that their combined width is not larger than the textwidth.

You can also pack a paragraph of your choice into a box with either the

`\parbox[`*pos*`]{`*width*`}{`*text*`}`

command or the

`\begin{minipage}[`*pos*`]{`*width*`}` text `\end{minipage}`

environment. The `pos` parameter can take one of the letters `c`, `t` or `b` to control the vertical alignment of the box, relative to the baseline of the surrounding text. `width` takes a length argument specifying the width of the box. The main difference between a `minipage` and a `parbox` is that you cannot use all commands and environments inside a `parbox` while almost anything is possible in a `minipage`.

While `\parbox` packs up a whole paragraph doing line breaking and everything, there is also a class of boxing commands which operates only on horizontally aligned material. We already know one of them. It's called `\mbox`, it simply packs up a series of boxes into another one, and can be used to prevent LaTeX from breaking two words. As you can put boxes inside boxes, these horizontal box packers give you ultimate flexibility.

`\makebox[`*width*`][`*pos*`]{`*text*`}`

`width` defines the width of the resulting box as seen from the outside.[5] Apart from the length expressions you can also use `\width`, `\height`, `\depth` and `\totalheight` in the width parameter. They are set from values obtained by measuring the typeset *text*. The *pos* parameter takes a one letter value: **c**enter, **l**eft flush, **r**ight flush or **s** which spreads the text inside the box to fill it.

The command `\framebox` works exactly the same as `\makebox`, but it draws a box around the text.

The following example shows you some things you could do with the `\makebox` and `\framebox` commands.

[5]This means it can be smaller than the material inside the box. You can even set the width to 0pt so that the text inside the box will be typeset without influencing the surrounding boxes.

```
\makebox[\textwidth]{%
    c e n t r a l}\par
\makebox[\textwidth][s]{%
    s p r e a d}\par
\framebox[1.1\width]{Guess I'm
    framed now!} \par
\framebox[0.8\width][r]{Bummer,
    I am too wide} \par
\framebox[1cm][l]{never
    mind, so am I}
Can you read this?
```

c e n t r a l

s p r e a d

Guess I'm framed now!

Bummer, I am too wide

never mind, so am I Can you read this?

Now that we control the horizontal, the obvious next step is to go for the vertical.[6] No problem for LaTeX. The

`\raisebox{`*lift*`}[`*depth*`][`*height*`]{`*text*`}`

command lets you define the vertical properties of a box. You can use `\width`, `\height`, `\depth` and `\totalheight` in the first three parameters, in order to act upon the size of the box inside the *text* argument.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}}
he shouted but not even the next
one in line noticed that something
terrible had happened to him.
```

Aaaaaaargh he shouted but not even the next one in line noticed that something terrible had happened to him.

5.7 Rules and Struts

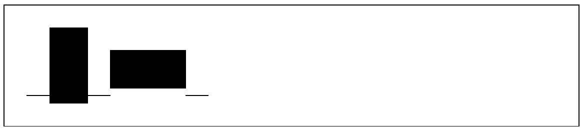
A few pages back you may have noticed the command

`\rule[`*lift*`]{`*width*`}{`*height*`}`

In normal use it produces a simple black box.

[6] Total control is only to be obtained by controlling both the horizontal and the vertical ...

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



This is useful for drawing vertical and horizontal lines. The line on the title page for example, has been created with a `\rule` command.

A special case is a rule with no width but a certain height. In professional typesetting, this is called a strut. It is used to guarantee that an element on a page has a certain minimal height. You could use it in a `tabular` environment to make sure a row has a certain minimum height.

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Pitprop \ldots\\
\hline
\rule{0pt}{4ex}Strut\\
\hline
\end{tabular}
```

Pitprop …
Strut

Bibliography

[1] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.

[2] Donald E. Knuth. *The TeXbook,* Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.

[3] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.

[4] Each LaTeX installation should provide a so-called *LaTeX Local Guide* which explains the things which are special to the local system. It should be contained in a file called `local.tex`. Unfortunately, some lazy sysops do not provide such a document. In this case, go and ask your local LaTeX guru for help.

[5] LaTeX3 Project Team. *LaTeX 2ε for authors.* Comes with the LaTeX 2ε distribution as `usrguide.tex`.

[6] LaTeX3 Project Team. *LaTeX 2ε for Class and Package writers.* Comes with the LaTeX 2ε distribution as `clsguide.tex`.

[7] LaTeX3 Project Team. *LaTeX 2ε Font selection.* Comes with the LaTeX 2ε distribution as `fntguide.tex`.

[8] D. P. Carlisle. *Packages in the 'graphics' bundle.* Comes with the 'graphics' bundle as `grfguide.tex`, available from the same source your LaTeX distribution came from.

[9] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of LaTeX's verbatim Environments.* Comes with the 'tools' bundle as `verbatim.dtx`, available from the same source your LaTeX distribution came from.

[10] Graham Williams. *The TeX Catalogue* is a very complete listing of many TeX and LaTeX related packages. Available online from `CTAN:/tex-archive/help/Catalogue/catalogue.html`

[11] Keith Reckdahl. *Using EPS Graphics in LaTeX 2ε Documents* which explains everything and much more than you ever wanted to know about EPS files and their use in LaTeX documents. Available online from `CTAN:/tex-archive/info/epslatex.ps`

Index

Symbols

\!, 45
", 21
$, 39
', 42
\(, 39
\), 39
\,, 40, 45
-, 22
−, 22
\-, 21
–, 22
—, 22
., space after, 26
. . . , 23
.aux, 13
.cls, 13
.dtx, 13
.dvi, 13
.idx, 13
.ilg, 13
.ind, 13
.ins, 13
.lof, 13
.log, 13
.lot, 13
.sty, 13
.tex, 13
.toc, 13
\:, 45
\;, 45
\@, 26
\[, 40
\\, 19, 31–33, 76
*, 19
\], 40
^, 42
_, 42
|, 34
~, 26

A

A4 paper, 11
A5 paper, 11
accent, 23
acute, 24
\addtolength, 78
advantages of LaTeX, 3
æ, 24
amsbsy, 50
amsfonts, 41, 58
amsmath, 43, 45, 46, 48, 50
amssymb, 41, 52
\and, 28
\appendix, 27, 28
\arccos, 43
\arcsin, 43
\arctan, 43
\arg, 43
array, 46
arrow symbols, 42
article class, 10
\atop, 43
\author, 28

B

B5 paper, 11
babel, 20, 24
\backmatter, 28
backslash, 6
\backslash, 5
base font size, 11

\begin, 30
\bibitem, 61
bibliography, 61
\Big, 45
\big, 45
\Bigg, 45
\bigg, 45
\bigskip, 76
\binom, 43
blackboard bold, 41
bm, 50
\bmod, 43
bold face, 71
bold symbols, 41, 50
\boldmath, 50
\boldsymbol, 50
book class, 10
braces, 44

C

calc, 78
\caption, 37, 38
\cdot, 43
\cdots, 45
center, 31
\chapter, 27
\chaptermark, 63
\choose, 43
\ci, 67
\cite, 61
\cleardoublepage, 38
\clearpage, 38
\cline, 34
coloured text, 10
comma, 23
commands, 6
- \!, 45
- \(, 39
- \), 39
- \,, 40, 45
- \-, 21
- \:, 45
- \;, 45
- \@, 26
- \[, 40
- \\, 19, 31–33, 76
- *, 19
- \], 40
- \addtolength, 78
- \and, 28
- \appendix, 27, 28
- \arccos, 43
- \arcsin, 43
- \arctan, 43
- \arg, 43
- \atop, 43
- \author, 28
- \backmatter, 28
- \backslash, 5
- \begin, 30
- \bibitem, 61
- \Big, 45
- \big, 45
- \Bigg, 45
- \bigg, 45
- \bigskip, 76
- \binom, 43
- \bmod, 43
- \boldmath, 50
- \boldsymbol, 50
- \caption, 37, 38
- \cdot, 43
- \cdots, 45
- \chapter, 27
- \chaptermark, 63
- \choose, 43
- \ci, 67
- \cite, 61
- \cleardoublepage, 38
- \clearpage, 38
- \cline, 34
- \cos, 43
- \cosh, 43
- \cot, 43
- \coth, 43
- \csc, 43
- \date, 28
- \ddots, 45

\deg, 43
\depth, 80, 81
\det, 43
\dim, 43
\displaystyle, 49
\documentclass, 9, 10, 13, 20
\dq, 26
\dum, 67
\emph, 30, 71
\end, 30
\exp, 43
\fbox, 21
\footnote, 29, 38
\footnotesize, 71
\frac, 43
\framebox, 80
\frenchspacing, 27
\frontmatter, 28
\fussy, 20
\gcd, 43
\genfrac, 43
\height, 80, 81
\hline, 33
\hom, 43
\hspace, 69, 75
\Huge, 71
\huge, 71
\hyphenation, 20
\idotsint, 46
\iiiint, 46
\iiint, 46
\iint, 46
\include, 14, 15
\includegraphics, 60, 79
\includeonly, 15
\indent, 75
\index, 62
\inf, 43
\input, 15
\int, 44
\item, 31
\ker, 43
\label, 29, 40
\LARGE, 71
\Large, 71
\large, 71
\LaTeX, 21
\LaTeXe, 21
\ldots, 23, 45
\left, 44
\leftmark, 63
\lg, 43
\lim, 43
\liminf, 43
\limsup, 43
\linebreak, 19
\linespread, 74
\listoffigures, 37
\listoftables, 37
\ln, 43
\log, 43
\mainmatter, 28
\makebox, 80
\makeindex, 62
\maketitle, 28
\mathbb, 41
\mathbf, 72
\mathcal, 72
\mathit, 72
\mathnormal, 72
\mathrm, 48, 72
\mathsf, 72
\mathtt, 72
\max, 43
\mbox, 21, 23, 80
\min, 43
\multicolumn, 35
\newcommand, 68, 69
\newenvironment, 69
\newline, 19
\newpage, 19
\newtheorem, 49
\noindent, 75
\nolinebreak, 19
\nonumber, 47
\nopagebreak, 19
\normalsize, 71
\overbrace, 42

\overleftarrow, 42
\overline, 42
\overrightarrow, 42
\pagebreak, 19
\pageref, 29
\pagestyle, 14
\par, 71
\paragraph, 27
\parbox, 80
\parindent, 74
\parskip, 74
\part, 27
\phantom, 38, 48
\pmod, 43
\Pr, 43
\printindex, 63
\prod, 44
\protect, 38
\providecommand, 69
\ProvidesPackage, 70
\qquad, 40, 45
\quad, 40, 45
\raisebox, 81
\ref, 29, 40
\renewcommand, 68
\renewenvironment, 69
\right, 44, 46
\right., 44
\rightmark, 63
\rule, 69, 81, 82
\scriptscriptstyle, 49
\scriptsize, 71
\scriptstyle, 49
\sec, 43
\section, 27, 38
\sectionmark, 63
\setlength, 74, 78
\settodepth, 79
\settoheight, 79
\settowidth, 79
\sin, 43
\sinh, 43
\sloppy, 20
\small, 71
\smallskip, 76
\sqrt, 42
\stackrel, 44
\stretch, 69, 75
\subparagraph, 27
\subsection, 27
\subsectionmark, 63
\subsubsection, 27
\sum, 44
\sup, 43
\tableofcontents, 27
\tan, 43
\tanh, 43
\TeX, 21
\textbf, 71
\textit, 71
\textmd, 71
\textnormal, 71
\textrm, 48, 71
\textsc, 71
\textsf, 71
\textsl, 71
\textstyle, 49
\texttt, 71
\textup, 71
\thispagestyle, 14
\tiny, 71
\title, 28
\tnss, 68
\today, 21
\totalheight, 80, 81
\underbrace, 42
\underline, 30, 42
\usepackage, 10, 13, 24, 25, 70
\vdots, 45
\vec, 42
\verb, 33
\verbatiminput, 64
\vspace, 76
\widehat, 42
\widetilde, 42
\width, 80, 81
comment, 7

comments, 6
\cos, 43
\cosh, 43
\cot, 43
\coth, 43
cross-references, 29
\csc, 43
curly braces, 6, 71

D
dash, 22
\date, 28
dcolumn, 34
\ddots, 45
decimal alignment, 34
\deg, 43
degree symbol, 22
delimiters, 44
\depth, 80, 81
description, 31
\det, 43
Deutsch, 26
diagonal dots, 45
\dim, 43
dimensions, 75
displaymath, 40
\displaystyle, 49
doc, 12
document font size, 11
document title, 11
\documentclass, 9, 10, 13, 20
dotless ı and ȷ, 24
double line spacing, 74
double sided, 11
\dq, 26
\dum, 67
dvips, 9

E
ellipsis, 23
em-dash, 22
\emph, 30, 71
empty, 14
en-dash, 22
Encapsulated PostScript, 59
\end, 30
enumerate, 31
environments
- array, 46
- center, 31
- comment, 7
- description, 31
- displaymath, 40
- enumerate, 31
- eqnarray, 47
- equation, 40
- figure, 35, 37
- flushleft, 31
- flushright, 31
- itemize, 31
- lscommand, 67
- math, 39
- minipage, 80
- parbox, 80
- quotation, 32
- quote, 32
- table, 35, 37
- tabular, 33, 79
- thebibliography, 61
- verbatim, 32, 64
- verse, 32

eqnarray, 47
equation, 40
equation system, 47
eucal, 58
eufrak, 58
executive paper, 11
\exp, 43
exponent, 42
exscale, 12, 45
extension, 12

F
fancyhdr, 63, 64
\fbox, 21
figure, 35, 37
file types, 12
floating bodies, 35

flushleft, 31
flushright, 31
foiltex, 10
font, 70
font encoding, 12
font size, 70, 71
fontenc, 12, 25
footer, 14
\footnote, 29, 38
\footnotesize, 71
formulae, 39
\frac, 43
fraction, 43
fragile commands, 38
\framebox, 80
\frenchspacing, 27
\frontmatter, 28
\fussy, 20

G

\gcd, 43
\genfrac, 43
geometry, 65
German, 24, 26
GhostScript, 59
graphics, 10, 59
graphicx, 59
grave, 24
Greek letters, 41
grouping, 71

H

header, 14
textttheadings, 14
\height, 80, 81
\hline, 33
\hom, 43
horizontal
    brace, 42
    dots, 45
    line, 42
    space, 75
\hspace, 69, 75
\Huge, 71
\huge, 71
hyphen, 22
hyphenat, 65
\hyphenation, 20

I

\idotsint, 46
ifthen, 12
\iiiint, 46
\iiint, 46
\iint, 46
\include, 14, 15
\includegraphics, 60, 79
\includeonly, 15
\indent, 75
indentfirst, 75
index, 62
\index, 62
\inf, 43
\input, 15
input file, 7
inputenc, 12, 25
\int, 44
integral operator, 44
international, 24
italic, 71
\item, 31
itemize, 31

K

\ker, 43
Knuth, Donald E., 1

L

\label, 29, 40
Lamport, Leslie, 1
language, 24
\LARGE, 71
\Large, 71
\large, 71
\LaTeX, 21
LaTeX 2_ε, 2
LaTeX 2.09, 2
LaTeX3, 2
LaTeX3, 4

\LaTeXe, 21
latexsym, 12
layout, 78
\ldots, 23, 45
\left, 44
left aligned, 31
\leftmark, 63
legal paper, 11
letter paper, 11
\lg, 43
ligature, 23
\lim, 43
\liminf, 43
\limsup, 43
line spacing, 74
linebreak, 19
\linebreak, 19
\linespread, 74
\listoffigures, 37
\listoftables, 37
\ln, 43
\log, 43
long equations, 47
longtabular, 35
lscommand, 67

M

\mainmatter, 28
\makebox, 80
makeidx, 12, 62
makeidx package, 62
\makeindex, 62
makeindex program, 62
\maketitle, 28
margins, 76
math, 39
math font size, 48
math spacing, 45
\mathbb, 41
\mathbf, 72
\mathcal, 72
mathematical
    accents, 42
    delimiter, 45
    functions, 43
    minus, 22
mathematics, 39
\mathit, 72
\mathnormal, 72
\mathrm, 48, 72
mathrsfs, 58
\mathsf, 72
\mathtt, 72
\max, 43
\mbox, 21, 23, 80
\min, 43
minipage, 80
minus sign, 22
Mittelbach, Frank, 2
modulo function, 43
\multicolumn, 35

N

\newcommand, 68, 69
\newenvironment, 69
\newline, 19
\newpage, 19
\newtheorem, 49
\noindent, 75
\nolinebreak, 19
\nonumber, 47
\nopagebreak, 19
\normalsize, 71

O

œ, 24
one column, 11
option, 10
optional parameters, 6
\overbrace, 42
overfull hbox, 20
\overleftarrow, 42
\overline, 42
\overrightarrow, 42

P

package, 7, 10, 67
packages
    amsbsy, 50

amsfonts, 41, 58
amsmath, 43, 45, 46, 48, 50
amssymb, 41, 52
babel, 20, 24
bm, 50
calc, 78
dcolumn, 34
doc, 12
eucal, 58
eufrak, 58
exscale, 12, 45
fancyhdr, 63, 64
fontenc, 12, 25
geometry, 65
graphicx, 59
hyphenat, 65
ifthen, 12
indentfirst, 75
inputenc, 12, 25
latexsym, 12
layout, 78
longtabular, 35
makeidx, 12, 62
mathrsfs, 58
showidx, 63
supertabular, 35
syntonly, 12, 15
verbatim, 7, 64
page layout, 76
page style, 14
empty, 14
headings, 14
plain, 14
\pagebreak, 19
\pageref, 29
\pagestyle, 14
paper size, 11, 76
\par, 71
paragraph, 17
\paragraph, 27
parameter, 6
\parbox, 80
parbox, 80
\parindent, 74
\parskip, 74
\part, 27
period, 23
\phantom, 38, 48
placement specifier, 36
plain, 14
\pmod, 43
PostScript, 59
\Pr, 43
preamble, 7
prime, 42
\printindex, 63
\prod, 44
product operator, 44
\protect, 38
\providecommand, 69
\ProvidesPackage, 70

Q

\qquad, 40, 45
\quad, 40, 45
quotation, 32
quotation marks, 21
quote, 32

R

\raisebox, 81
\ref, 29, 40
\renewcommand, 68
\renewenvironment, 69
report class, 10
reserved characters, 5
\right, 44, 46
right-aligned, 31
\right., 44
\rightmark, 63
roman, 71
\rule, 69, 81, 82

S

sans serif, 71
Scandinavian letters, 24
\scriptscriptstyle, 49
\scriptsize, 71
\scriptstyle, 49

\sec, 43
\section, 27, 38
\sectionmark, 63
\setlength, 74, 78
\settodepth, 79
\settoheight, 79
\settowidth, 79
showidx, 63
\sin, 43
single sided, 11
\sinh, 43
slanted, 71
slides class, 10
\sloppy, 20
\small, 71
small caps, 71
\smallskip, 76
space, 5
special character, 23
\sqrt, 42
square brackets, 6
square root, 42
\stackrel, 44
\stretch, 69, 75
structure, 7
strut, 82
\subparagraph, 27
subscript, 42
\subsection, 27
\subsectionmark, 63
\subsubsection, 27
\sum, 44
sum operator, 44
\sup, 43
supertabular, 35
syntonly, 12, 15

T

table, 33
table, 35, 37
table of contents, 27
\tableofcontents, 27
tabular, 33, 79
\tan, 43
\tanh, 43
\TeX, 21
\textbf, 71
\textit, 71
\textmd, 71
\textnormal, 71
\textrm, 48, 71
\textsc, 71
\textsf, 71
\textsl, 71
\textstyle, 49
\texttt, 71
\textup, 71
thebibliography, 61
\thispagestyle, 14
three dots, 45
tilde, 22, 42
tilde (~), 26
\tiny, 71
title, 11, 28
\title, 28
\tnss, 68
\today, 21
\totalheight, 80, 81
two column, 11

U

umlaut, 24
\underbrace, 42
underfull hbox, 20
\underline, 30, 42
units, 75, 76
upright, 71
URL, 22
\usepackage, 10, 13, 24, 25, 70

V

\vdots, 45
\vec, 42
vectors, 42
\verb, 33
verbatim, 7, 64
verbatim, 32, 64
\verbatiminput, 64

`verse`, 32
vertical dots, 45
vertical space, 76
`\vspace`, 76

W

whitespace, 5
  after commands, 6
  at the start of a line, 5
`\widehat`, 42
`\widetilde`, 42
`\width`, 80, 81
www, 22
WYSIWYG, 3

X

xdvi, 9