LaTeX

en.wikibooks.org

June 6, 2015

On the 28th of April 2012 the contents of the English as well as German Wikibooks and Wikipedia projects were licensed under Creative Commons Attribution-ShareAlike 3.0 Unported license. A URI to this license is given in the list of figures on page 695. If this document is a derived work from the contents of one of these projects and the content was still licensed by the project under this license at the time of derivation this document has to be licensed under the same, a similar or a compatible license, as stated in section 4b of the license. The list of contributors is included in chapter Contributors on page 685. The licenses GPL, LGPL and GFDL are included in chapter Licenses on page 707, since this book and/or parts of it may or may not be licensed under one or more of these licenses, and thus require inclusion of these licenses. The licenses of the figures are given in the list of figures on page 695. This PDF was generated by the LaTeX typesetting software. The LaTeX source code is included as an attachment (`source.7z.txt`) in this PDF file. To extract the source from the PDF file, you can use the `pdfdetach` tool including in the `poppler` suite, or the `http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/` utility. Some PDF viewers may also let you save the attachment to a file. After extracting it from the PDF file you have to rename it to `source.7z`. To uncompress the resulting archive we recommend the use of `http://www.7-zip.org/`. The LaTeX source itself was generated by a program written by Dirk Hünniger, which is freely available under an open source license from `http://de.wikibooks.org/wiki/Benutzer:Dirk_Huenniger/wb2pdf`.

Contents

Part I.

Getting Started

1. Introduction

1.1. What is TeX?

TeX[1] is a low-level markup and programming language created by Donald Knuth[2] to typeset documents attractively and consistently. 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. With the release of 8-bit character support in 1989, TeX development has been essentially frozen with only bug fixes released periodically. TeX is a programming language in the sense that it supports the if-else construct: you can make calculations with it (that are performed while compiling the document), etc., but you would find it very hard to do anything else but typesetting with it. The fine control TeX offers over document structure and formatting makes it a powerful—and formidable—tool. TeX is renowned for being extremely stable, for running on many different kinds of computers, and for being virtually bug free. The version numbers of TeX are converging toward π, with a current version number of 3.1415926.

The name TeX is intended by its developer to be /'tɛx/ , with the final consonant of *loch* or *Bach* . (Donald E. Knuth, *The TeXbook*) The letters of the name are meant to represent the capital Greek letters tau, epsilon, and chi, as TeX is an abbreviation of *τέχνη* (TEXNH – technē), Greek for both "art" and "craft", which is also the root word of *technical* . English speakers often pronounce it /'tɛk/ , like the first syllable of *technical* .

Programming in TeX generally progresses along a very gradual learning curve, requiring a significant investment of time to build custom macros for text formatting. Fortunately, document preparation systems based on TeX, consisting of collections of pre-built macros, do exist. These pre-built macros are time saving, and automate certain repetitive tasks and help reduce user introduced errors; however, this convenience comes at the cost of complete design flexibility. One of the most popular macro packages is called **LaTeX** .

1.2. What is LaTeX?

LaTeX (pronounced either "Lah-tech" or "Lay-tech") is a macro package based on TeX created by Leslie Lamport[3]. Its purpose is to simplify TeX typesetting, especially for

1 `http://en.wikibooks.org/wiki/TeX`
2 `http://en.wikipedia.org/wiki/Donald%20Knuth`
3 `http://en.wikipedia.org/wiki/Leslie%20Lamport`

documents containing mathematical formulae. Within the typesetting system, its name is formatted as LaTeX.

Many later authors have contributed extensions, called *packages* or *styles* , to LaTeX. Some of these are bundled with most TeX/LaTeX software distributions; more can be found in the Comprehensive TeX Archive Network (CTAN[4]).

Since LaTeX comprises a group of TeX[5] commands, LaTeX document processing is essentially programming. You create a text file in LaTeX markup, which LaTeX reads to produce the final document.

This approach has some disadvantages in comparison with a WYSIWYG[6] (What You See Is What You Get) program such as Openoffice.org[7] Writer or Microsoft Word[8].

In LaTeX:

- You don't (usually) see the final version of the document when editing it.
- You generally need to know the necessary commands for LaTeX markup.
- It can sometimes be difficult to obtain a certain look for the document.

On the other hand, there are certain advantages to the LaTeX approach:

- Document sources can be read with any text editor and understood, unlike the complex binary and XML[9] formats used with WYSIWYG programs.
- You can concentrate purely on the structure and contents of the document, not get caught up with superficial layout issues.
- You don't need to manually adjust fonts, text sizes, line heights, or text flow for readability, as LaTeX takes care of them automatically.
- In LaTeX the document structure is visible to the user, and can be easily copied to another document. In WYSIWYG applications it is often not obvious how a certain formatting was produced, and it might be impossible to copy it directly for use in another document.
- The layout, fonts, tables and so on are consistent throughout the document.
- Mathematical formulae can be easily typeset.
- Indexes, footnotes, citations and references are generated easily.
- Since the document source is plain text, tables, figures, equations, etc. can be generated programmatically with any language.
- You are forced to structure your documents correctly.

The LaTeX-like approach can be called WYSIWYM[10], i.e. What You See Is What You Mean: you can't see what the final version will look like while typing. Instead you see the logical structure of the document. LaTeX takes care of the formatting for you.

The LaTeX document is a plain text file containing the content of the document, with additional markup. When the source file is processed by the macro package, it can produce

4 `http://www.ctan.org`
5 `http://en.wikibooks.org/wiki/TeX`
6 `http://en.wikipedia.org/wiki/WYSIWYG`
7 `http://en.wikipedia.org/wiki/Openoffice.org`
8 `http://en.wikipedia.org/wiki/Microsoft%20Word`
9 `http://en.wikipedia.org/wiki/XML`
10 `http://en.wikipedia.org/wiki/WYSIWYM`

documents in several formats. LaTeX natively supports DVI[11] and PDF, but by using other software you can easily create PostScript, PNG, JPEG, etc.

1.3. Philosophy of use

1.3.1. Flexibility and modularity

One of the most frustrating things beginners and even advanced users might encounter using LaTeX is the lack of flexibility regarding the document design and layout. If you want to design your document in a very specific way, you may have trouble accomplishing this. Keep in mind that LaTeX *does* the formatting for you, and mostly the right way. If it is not exactly what you desired, then the LaTeX way is at least not worse, if not better. One way to look at it is that LaTeX is a bundle of macros for TeX that aims to carry out everything regarding document formatting, so that the writer only needs to care about content. If you really want flexibility, use plain TeX instead.

One solution to this dilemma is to make use of the modular possibilities of LaTeX. You can build your own macros, or use macros developed by others. You are likely not the first person to face some particular formatting problem, and someone who encountered a similar problem before may have published their solution as a package.

CTAN[12] is a good place to find many resources regarding TeX and derivative packages. It is the first place where you should begin searching.

1.3.2. Questions and documentation

Besides internet resources being plentiful, the best documentation source remains the official manual for every specific package, and the reference documentation, i.e., the *TeXbook* by D. Knuth and *LaTeX: A document preparation system* by L. Lamport.

Therefore before rushing on your favorite web search engine, we really urge you to have a look at the package documentation that causes troubles. This official documentation is most commonly installed along your TeX distribution, or may be found on CTAN[13].

1.4. Terms regarding TeX

Document preparation systems

LaTeX is a document preparation system based on TeX. So the system is the combination of the language and the macros.

Distributions

11 `http://en.wikipedia.org/wiki/DVI%20file%20format`
12 `http://www.ctan.org/`
13 `http://www.ctan.org/`

TeX distributions are collections of packages and programs (compilers, fonts, and macro packages) that enable you to typeset without having to manually fetch files and configure things.

Engines

An engine is an executable that can turn your source code into a printable output format. The engine by itself only handles the syntax, it also needs to load fonts and macros to fully understand the source code and generate output properly. The engine will determine what kind of source code it can read, and what format it can output (usually DVI or PDF).

All in all, distributions are an easy way to install what you need to use the engines and the systems you want. Distributions usually target specific operating systems. You can use different systems on different engines, but sometimes there are restrictions. Code written for TeX, LaTeX or ConTeXt are (mostly) not compatible. Additionally, engine-specific code (like font for XeTeX) may not be compiled by every engine.

When searching for information on LaTeX, you might also stumble upon XeTeX[14], ConTeXt[15], LuaTeX[16] or other names with a -TeX suffix. Let's recap most of the terms in this table.

Systems	**Descriptions**
ConTeXt	A TeX-based document preparation system (as LaTeX is) with a very consistent and easy syntax and support for pdfTeX, XeTeX and LuaTeX engines.It does not have the same objective as LaTeX however.
LaTeX	A TeX-based document preparation system designed by Leslie Lamport. It is actually a set of macros for TeX. It aims at taking care of the formatting process.
MetaFont	A high-quality font system designed by Donald Knuth along TeX.
MetaPost	A descriptive vector graphics language based on MetaFont.
TeX	The original language designed by Donald Knuth.

Engines	**Descriptions**
`luatex` , `lualatex`	A TeX engine with Lua scripting engine embedded aiming at making TeX internals more flexible.
`pdftex` , `pdflatex`	The engines (PDF compilers).
`tex` , `latex`	The engines (DVI compilers).
`xetex` , `xelatex`	a TeX engine which uses Unicode and supports widely popular `.ttf` and `.otf` fonts. See Fonts[17].

TeX Distributions	**Descriptions**
MacTeX	A TeX Live based distribution targetting Mac OS X.
MiKTeX	A TeX distribution for Windows.
TeX Live	A cross-platform TeX distribution.

14 `http://en.wikipedia.org/wiki/XeTeX`
15 `http://en.wikipedia.org/wiki/ConTeXt`
16 `http://en.wikipedia.org/wiki/LuaTeX`
17 Chapter 9 on page 95

1.5. What next?

In the next chapter we will proceed to the installation[18]. Then we will compile our first LaTeX file[19].

Throughout this book you should also utilise other means for learning about LaTeX. Good sources are:

- the `#latex`[20] IRC channel on Freenode,
- the TeX Stack Exchange[21] Q&A,
- the TeX[22] FAQ,
- and the TeXample.net[23] Community.

de:LaTeX/_Einleitung[24]

18 Chapter 2 on page 11
19 Chapter 4 on page 37
20 `http://webchat.freenode.net?channels=latex`
21 `http://tex.stackexchange.com/`
22 `http://www.tex.ac.uk/cgi-bin/texfaq2html`
23 `http://www.texample.net/`
24 `http://de.wikibooks.org/wiki/LaTeX%2F_Einleitung`

2. Installation

If this is the first time you are trying out LaTeX, you don't even need to install anything. For quick testing purpose you may just create a user account with an online LaTeX editor[1] and continue this tutorial in the next chapter. These websites offer collaboration capabilities while allowing you to experiment with LaTeX syntax without having to bother with installing and configuring a distribution and an editor. When you later feel that you would benefit from having a standalone LaTeX installation, you can return to this chapter and follow the instructions below.

LaTeX is not a program by itself; it is a language. *Using LaTeX* requires a bunch of tools. Acquiring them manually would result in downloading and installing multiple programs in order to have a suitable computer system that can be used to create LaTeX output, such as PDFs. **TeX Distributions** help the user in this way, in that it is a single step installation process that provides (almost) everything.

At a minimum, you'll need a TeX distribution, a good text editor and a DVI or PDF viewer. More specifically, the basic requirement is to have a TeX compiler (which is used to generate output files from source), fonts, and the LaTeX macro set. Optional, and recommended installations include an attractive editor to write LaTeX source documents (this is probably where you will spend most of your time), and a bibliographic management program to manage references if you use them a lot.

2.1. Distributions

TeX and LaTeX are available for most computer platforms, since they were programmed to be very portable. They are most commonly installed using a distribution, such as teTeX, MiKTeX, or MacTeX. TeX distributions are collections of packages and programs (compilers, fonts, and macro packages) that enable you to typeset without having to manually fetch files and configure things. LaTeX is just a set of macro packages built for TeX.

The recommended distributions for each of the major operating systems are:

- TeX Live[2] is a major TeX distribution for *BSD, GNU/Linux, Mac OS X and Windows.
- MiKTeX[3] is a Windows-specific distribution.
- MacTeX[4] is a Mac OS-specific distribution based on TeX Live.

1 Chapter 2.3.5 on page 23
2 `http://www.tug.org/texlive/`
3 `http://www.miktex.org/`
4 `http://www.tug.org/mactex/`

These, however, do not necessarily include an editor. You might be interested in other programs that are not part of the distribution, which will help you in writing and preparing TeX and LaTeX files.

2.1.1. *BSD and GNU/Linux

In the past, the most common distribution used to be **teTeX** . As of May 2006 teTeX is no longer actively maintained and its former maintainer Thomas Esser recommended TeX Live as the replacement.[5]

The easy way to get TeX Live is to use the package manager or portage tree coming with your operating system. Usually it comes as several packages, with some of them being essential, other optional. The *core* TeX Live packages should be around 200-300 MB.

If your *BSD or GNU/Linux distribution does not have the TeX Live packages, you should report a wish to the bug tracking system. In that case you will need to download TeX Live[6] yourself and run the installer by hand.

You may wish to install the content of TeX Live more selectively. See below[7].

2.1.2. Mac OS X

Mac OS X users may use MacTeX[8], a TeX Live-based distribution supporting TeX, LaTeX, AMSTeX, ConTeXt, XeTeX and many other core packages. Download MacTeX.mpkg.zip on the MacTeX page[9], unzip it and follow the instructions. Further information for Mac OS X users can be found on the TeX on Mac OS X Wiki[10].

Since Mac OS X is also a Unix-based system, TeX Live is naturally available through MacPorts[11] and Fink[12]. Homebrew[13] users should use the official MacTeX installer[14] because of the unique directory structure used by TeX Live[15]. Further information for Mac OS X users can be found on the TeX on Mac OS X Wiki[16].

2.1.3. Microsoft Windows

Microsoft Windows users can install MiKTeX[17] onto their computer. It has an easy installer that takes care of setting up the environment and downloading core packages. This

5 teTeX Home Page ^{`http://www.tug.org/tetex/`} (Retrieved January 31, 2007)
6 `http://www.tug.org/texlive/acquire.html`
7 Chapter 2.2 on page 13
8 `http://tug.org/mactex/`
9 `http://www.tug.org/mactex/`
10 `http://mactex-wiki.tug.org/`
11 `http://www.macports.org/`
12 `http://www.finkproject.org/`
13 `http://brew.sh/`
14 `http://www.tug.org/mactex/`
15 `https://github.com/Homebrew/homebrew/issues/1087`
16 `http://mactex-wiki.tug.org/`
17 `http://miktex.org/`

distribution has advanced features, such as automatic installation of packages, and simple interfaces to modify settings, such as default paper sizes.

There is also a port of TeX Live available for Windows.

2.2. Custom installation with TeX Live

This section targets users who want fine-grained control over their TeX distribution, like an installation with a minimum of disk space usage. If it is none of your concern, you may want to jump to the next section[18].

Picky users may wish to have more control over their installation. Common distributions might be tedious for the user caring about disk space. In fact, MikTeX and MacTeX and packaged TeX Live features hundreds of LaTeX packages, most of them which you will never use. Most Unix with a package manager will offer TeX Live as a set of several big packages, and you often have to install 300–400 MB for a functional system.

TeX Live features a manual installation with a lot of possible customizations. You can get the network installer at tug.org[19]. This installer allows you to select precisely the packages you want to install. As a result, you may have everything you need for less than 100 MB. TeX Live is then managed through its own package manager, *tlmgr* . It will let you configure the distributions, install or remove extra packages and so on.

You will need a Unix-based operating system for the following. Mac OS X, GNU/Linux or *BSD are fine. It may work for Windows but the process must be quite different.

TeX Live groups features and packages into different concepts:

- *Collections* are groups of packages that can always be installed individually, except for the *Essential programs and files* collection. You can install collections at any time.
- *Installation Schemes* group collections and packages. Schemes can only be used at installation time. You can select only one scheme at a time.

2.2.1. Minimal installation

We will give you general guidelines to install a minimal TeX distribution (*i.e.* , only for plain TeX).

1. Download the installer at `http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz` and extract it to a temporary folder.
2. Open a terminal in the extracted folder and log in as root.
3. Change the umask[20] permissions to 022 (user read/write/execute, group/others read/execute only) to make sure other users will have read-only access to the installed distribution.

18 Chapter 2.3 on page 17
19 `http://www.tug.org/texlive/acquire-netinstall.html`
20 `http://en.wikipedia.org/wiki/umask`

```
# umask 022
```

⚠ **Warning**

All administration operations for TeX Live should be made with a 022 umask. Otherwise you will not be able to use TeX at all with an unprivileged user.

1. Launch `install-tl` .
2. Select the *minimal scheme (plain only)* .
3. You may want to change the directory options. For example you may want to hide your personal macro folder which is located at TEXMFHOME. It is `~/texmf` by default. Replace it by `~/.texmf` to hide it.
4. Now the options:
   a) **use letter size instead of A4 by default:** mostly for users from the USA.
   b) **execution of restricted list of programs:** it is recommended to select it for security reasons. Otherwise it allows the TeX engines to call any external program. You may still configure the list afterwards.
   c) **create format files:** targetting a minimal disk space, the best choice depends on whether there is only one user on the system, then deselecting it is better, otherwise select it. From the help menu: *"If this option is set, format files are created for system-wide use by the installer. Otherwise they will be created automatically when needed. In the latter case format files are stored in user's directory trees and in some cases have to be re-created when new packages are installed."*
   d) **install font/macro doc tree:** useful if you are a developer, but very space consuming. Turn it off if you want to save space.
   e) **install font/macro source tree:** same as above.
   f) Symlinks are fine by default, change it if you know what you are doing.
5. Select portable installation if you install the distribution to an optical disc, or any kind of external media. Leave to default for a traditional installation on the system hard drive.

At this point it should display

```
1 collections out of 85, disk space required: 40 MB
```

or a similar space usage.

You can now proceed to installation: *start installation to hard disk* .

Don't forget to add the binaries to your PATH[21] as it's noticed at the end of the installation procedure.

21 `http://en.wikipedia.org/wiki/PATH%20%28variable%29`

2.2.2. First test

In a terminal write

```
$ tex '\empty Hello world!\bye'
$ pdftex '\empty Hello world!\bye'
```

You should get a DVI or a PDF file accordingly.

2.2.3. Configuration

Formerly, TeX distributions used to be configured with the `texconfig` tool from the teTeX distribution. TeX Live still features this tool, but recommends using its own tool instead: `tlmgr` . Note that as of January 2013 not all `texconfig` features are implemented by `tlmgr` . Only use `texconfig` when you cannot do what you want with `tlmgr` .

List current installation options:

```
tlmgr option
```

You can change the install options:

```
tlmgr option src 1
tlmgr option doc 0
tlmgr option paper letter
```

See the `TLMGR(1)` man page for more details on its usage. If you did not install the documents as told previously, you can still access the `tlmgr` man page with

```
tlmgr help
```

2.2.4. Installing LaTeX

⚠ **Warning**

Do not forget to set the root umask to 022 for all TeX Live administration operations.

Now we have a running plain TeX environment, let's install the base packages for LaTeX.

```
# tlmgr install latex latex-bin latexconfig latex-fonts
```

In this case you can omit `latexconfig latex-fonts` as they are auto-resolved dependencies to LaTeX. Note that `tlmgr` resolves some dependencies, but not all. You may need to install dependencies manually. Thankfully this is rarely too cumbersome.

Other interesting packages:

```
# tlmgr install amsmath babel carlisle ec geometry graphics hyperref lm
marvosym oberdiek parskip pdftex-def url
```

`amsmath`	The essentials for math typesetting.
`babel`	Internationalization support.
`carlisle`	Bundle package required for some `babel` features.
`ec`	Required for T1 encoding.
`geometry`	For page layout.
`graphics`	The essentials to import graphics.
`hyperref`	PDF bookmarks, PDF followable links, link style, TOC links, etc.
`lm`	One of the best Computer Modern style font available for several font encodings (such as T1).
`marvosym`	Several symbols, such as the official euro.
`oberdiek`	Bundle package required for some `geometry` features.
`parskip`	Let you configure paragraph breaks and indents properly.
`pdftex-def`	Required for some `graphics` features.
`url`	Required for some `hyperref` features.

If you installed a package you do not need anymore, use

```
# tlmgr remove <package>
```

2.2.5. Hyphenation

If you are using Babel for non-English documents, you need to install the hyphenation patterns for every language you are going to use. They are all packaged individually. For instance, use

```
# tlmgr install hyphen-{finnish,sanskrit}
```

for finnish and sanskrit hyphenation patterns.

Note that if you have been using another TeX distribution beforehand, you may still have hyphenation cache stored in you home folder. You need to remove it so that the new packages are taken into account. The TeX Live cache is usually stored in the `~/.texliveYYYY` folder (`YYYY` stands for the year). You may safely remove this folder as it contains only generated data. TeX compilers will re-generate the cache accordingly on next compilation.

2.2.6. Uninstallation

By default TeX Live will install in `/usr/local/texlive` . The distribution is quite proper as it will not write any file outside its folder, except for the cache (like font cache, hyphenation patters, etc.). By default,

- the system cache goes in `/var/lib/texmf` ;
- the user cache goes in `~/.texliveYYYY` .

Therefore TeX Live can be installed and uninstalled safely by removing the aforementioned folders.

Still, TeX Live provides a more convenient way to do this:

```
# tlmgr uninstall
```

You may still have to wipe out the folders if you put untracked files in them.

2.3. Editors

TeX and LaTeX source documents (as well as related files) are all text files, and can be opened and modified in almost any text editor. You should use a text editor (e.g. Notepad), not a word processor (Word, OpenOffice). Dedicated LaTeX editors are more useful than generic plain text editors, because they usually have autocompletion of commands, spell and error checking and handy macros.

2.3.1. Cross-platform

BaKoMa TeX

BaKoMa TeX[22] is an editor for Windows and Mac OS with WYSIWYG-like features. It takes care of compiling the LaTeX source and updating it constantly to view changes to document almost in real time. You can take an evaluation copy for 28 days.

Emacs

Emacs[23] is a general purpose, extensible text processing system. Advanced users can program it (in elisp) to make Emacs the best LaTeX environment that will fit their needs. In turn beginners may prefer using it in combination with AUCTeX[24] and Reftex (extensions that may be installed into the Emacs program). Depending on your configuration, Emacs can provide a complete LaTeX editing environment with auto-completion, spell-checking,

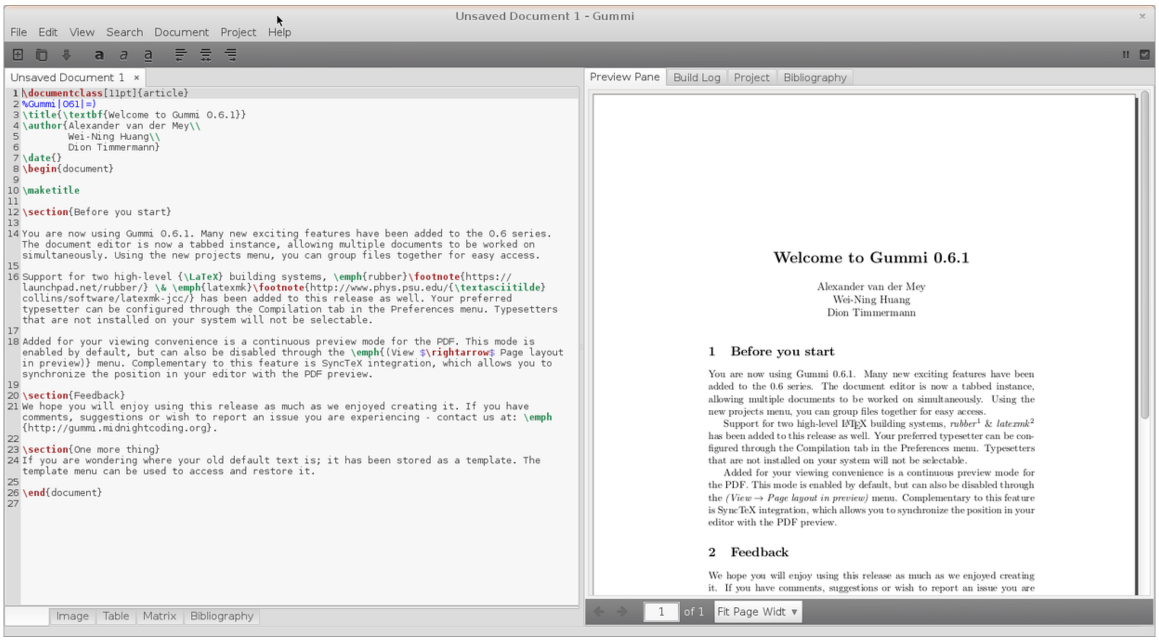
22 `http://bakoma-tex.com/menu/about.php`
23 `http://www.gnu.org/software/emacs`
24 `http://www.gnu.org/software/auctex/`

a complete set of keyboard shortcuts, table of contents view, document preview and many other features.

gedit-latex-plugin

Gedit with gedit-latex-plugin[25] is also worth trying out for users of GNOME. GEdit is a cross-platform application for Windows, Mac, and Linux

Gummi



```
\documentclass[11pt]{article}
%Gummi|061|=)
\title{\textbf{Welcome to Gummi 0.6.1}}
\author{Alexander van der Mey\\
        Wei-Ning Huang\\
        Dion Timmermann}
\date{}
\begin{document}

\maketitle

\section{Before you start}

You are now using Gummi 0.6.1. Many new exciting features have been added to the 0.6 series.
The document editor is now a tabbed instance, allowing multiple documents to be worked on
simultaneously. Using the new projects menu, you can group files together for easy access.

Support for two high-level {\LaTeX} building systems, \emph{rubber}\footnote{https://
launchpad.net/rubber/} \& \emph{latexmk}\footnote{http://www.phys.psu.edu/{\textasciitilde}
collins/software/latexmk-jcc/} has been added to this release as well. Your preferred
typesetter can be configured through the Compilation tab in the Preferences menu. Typesetters
that are not installed on your system will not be selectable.

Added for your viewing convenience is a continuous preview mode for the PDF. This mode is
enabled by default, but can also be disabled through the \emph{(View $\rightarrow$ Page layout
in preview)} menu. Complementary to this feature is SyncTeX integration, which allows you to
synchronize the position in your editor with the PDF preview.

\section{Feedback}
We hope you will enjoy using this release as much as we enjoyed creating it. If you have
comments, suggestions or wish to report an issue you are experiencing - contact us at: \emph
{http://gummi.midnightcoding.org}.

\section{One more thing}
If you are wondering where your old default text is; it has been stored as a template. The
template menu can be used to access and restore it.

\end{document}
```

Welcome to Gummi 0.6.1

Alexander van der Mey
Wei-Ning Huang
Dion Timmermann

1 Before you start

You are now using Gummi 0.6.1. Many new exciting features have been added to the 0.6 series. The document editor is now a tabbed instance, allowing multiple documents to be worked on simultaneously. Using the new projects menu, you can group files together for easy access.

Support for two high-level LaTeX building systems, *rubber*[1] & *latexmk*[2] has been added to this release as well. Your preferred typesetter can be configured through the Compilation tab in the Preferences menu. Typesetters that are not installed on your system will not be selectable.

Added for your viewing convenience is a continuous preview mode for the PDF. This mode is enabled by default, but can also be disabled through the *(View → Page layout in preview)* menu. Complementary to this feature is SyncTeX integration, which allows you to synchronize the position in your editor with the PDF preview.

2 Feedback

We hope you will enjoy using this release as much as we enjoyed creating it. If you have comments, suggestions or wish to report an issue you are

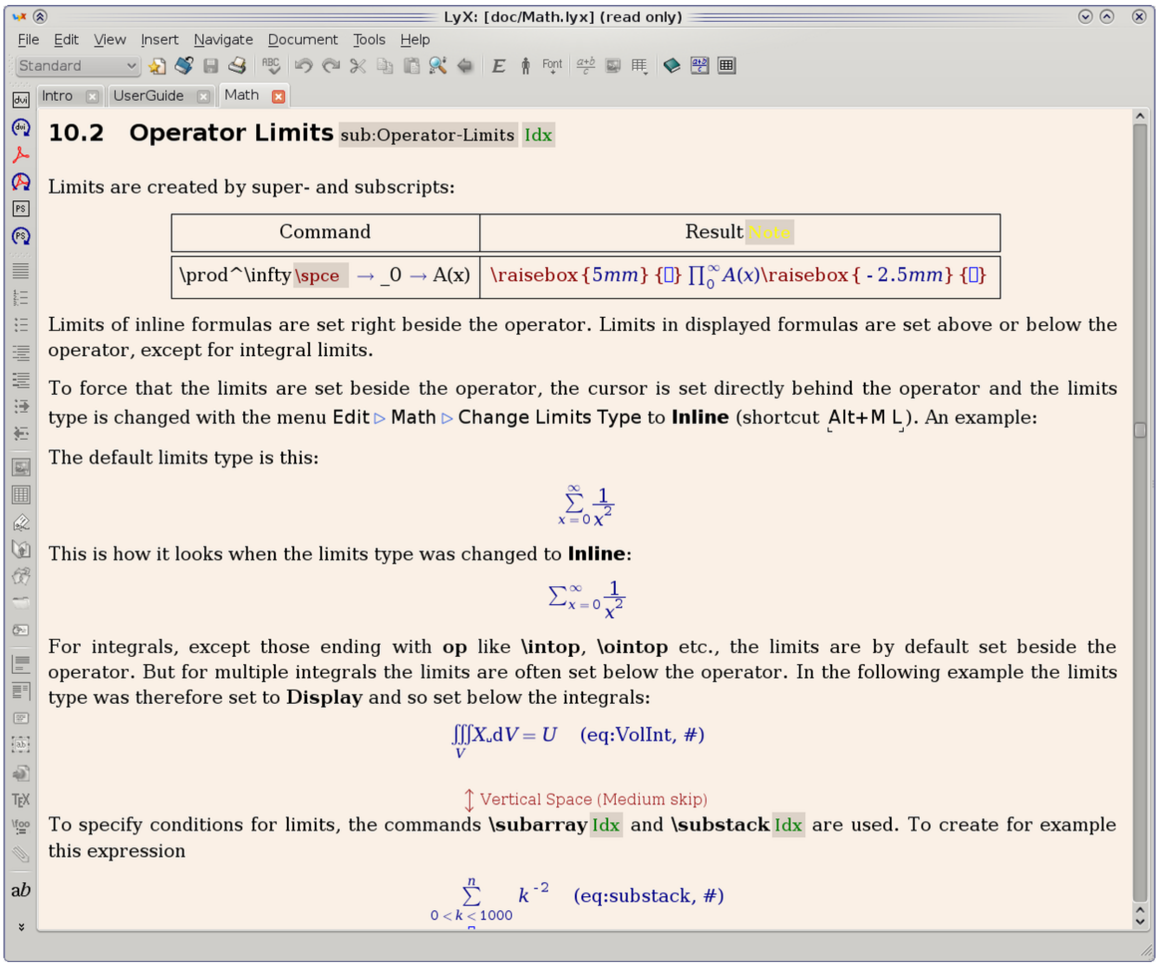
Figure 1 Screenshot of Gummi[a].

a http://en.wikipedia.org/wiki/Gummi%20%28software%29

Gummi[26] is a LaTeX editor for Linux, which compiles the output of pdflatex in realtime and shows it on the right half of the screen[27].

25 https://wiki.gnome.org/Apps/Gedit/LaTeXPlugin
26 http://en.wikipedia.org/wiki/Gummi%20%28software%29
27 Gummi ^{http://gummi.midnightcoding.org/}

LyX



LyX: [doc/Math.lyx] (read only)
File Edit View Insert Navigate Document Tools Help
Standard
Intro
UserGuide
Math

10.2 Operator Limits sub:Operator-Limits Idx

Limits are created by super- and subscripts:

Command	Result Note
\prod^\infty \spce → _0 → A(x)	\raisebox{5*mm*} {□} $\prod_0^\infty A(x)$\raisebox{ - 2.5*mm*} {□}

Limits of inline formulas are set right beside the operator. Limits in displayed formulas are set above or below the operator, except for integral limits.

To force that the limits are set beside the operator, the cursor is set directly behind the operator and the limits type is changed with the menu Edit ▷ Math ▷ Change Limits Type to **Inline** (shortcut Alt+M L). An example:

The default limits type is this:

$$\sum_{x=0}^{\infty}\frac{1}{x^2}$$

This is how it looks when the limits type was changed to **Inline**:

$$\sum\nolimits_{x=0}^{\infty}\frac{1}{x^2}$$

For integrals, except those ending with **op** like **\intop**, **\ointop** etc., the limits are by default set beside the operator. But for multiple integrals the limits are often set below the operator. In the following example the limits type was therefore set to **Display** and so set below the integrals:

$$\iiint\limits_V X_n dV = U \quad \text{(eq:VolInt, \#)}$$

Vertical Space (Medium skip)

To specify conditions for limits, the commands **\subarray** Idx and **\substack** Idx are used. To create for example this expression

$$\sum_{0<k<1000}^{n} k^{-2} \quad \text{(eq:substack, \#)}$$

Figure 2 LyX1.6.3

LyX[28] is a popular LaTeX editor for Windows, Linux and Mac OS. It contains formula and table editors and shows visual clues of the final document on the screen enabling users to write LaTeX documents without worrying about the actual syntax[29].

TeXmaker

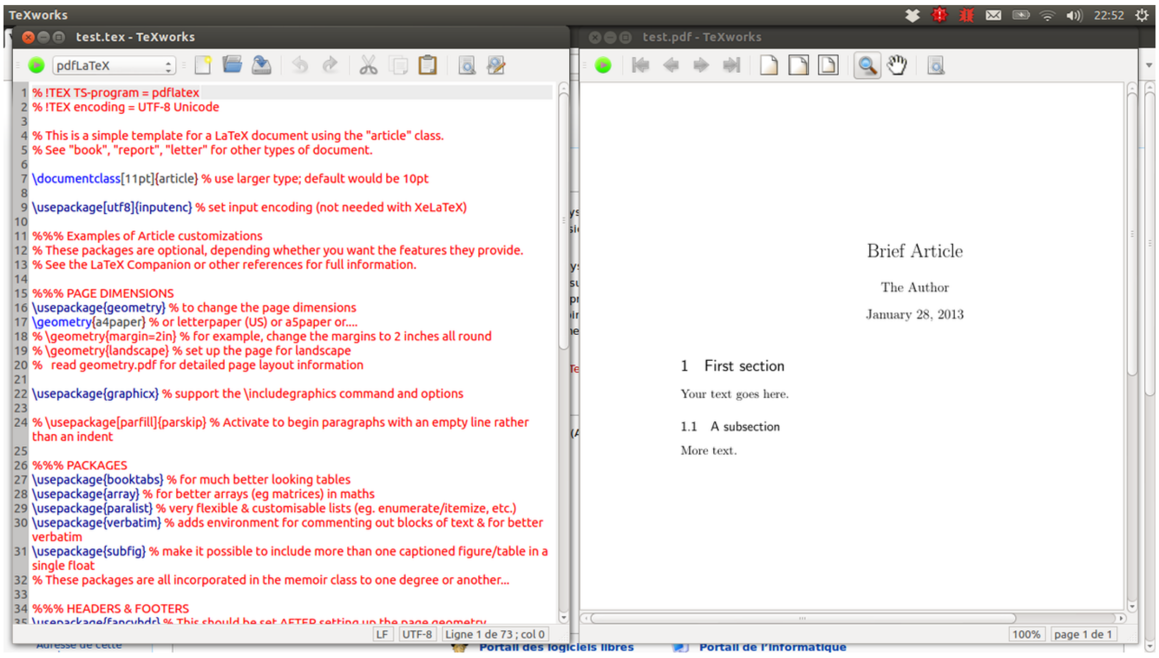
TeXmaker[30] is a cross-platform editor very similar to Kile in features and user interface. In addition it has its own PDF viewer.

28 `http://en.wikipedia.org/wiki/LyX`
29 LyX ^{`http://www.lyx.org/`}
30 `http://www.xm1math.net/texmaker/`

TeXstudio

TeXstudio[31] is a cross-platform open source LaTeX editor forked from Texmaker.

TeXworks



```
% !TEX TS-program = pdflatex
% !TEX encoding = UTF-8 Unicode

% This is a simple template for a LaTeX document using the "article" class.
% See "book", "report", "letter" for other types of document.

\documentclass[11pt]{article} % use larger type; default would be 10pt

\usepackage[utf8]{inputenc} % set input encoding (not needed with XeLaTeX)

%%% Examples of Article customizations
% These packages are optional, depending whether you want the features they provide.
% See the LaTeX Companion or other references for full information.

%%% PAGE DIMENSIONS
\usepackage{geometry} % to change the page dimensions
\geometry{a4paper} % or letterpaper (US) or a5paper or....
% \geometry{margin=2in} % for example, change the margins to 2 inches all round
% \geometry{landscape} % set up the page for landscape
%   read geometry.pdf for detailed page layout information

\usepackage{graphicx} % support the \includegraphics command and options

% \usepackage[parfill]{parskip} % Activate to begin paragraphs with an empty line rather than an indent

%%% PACKAGES
\usepackage{booktabs} % for much better looking tables
\usepackage{array} % for better arrays (eg matrices) in maths
\usepackage{paralist} % very flexible & customisable lists (eg. enumerate/itemize, etc.)
\usepackage{verbatim} % adds environment for commenting out blocks of text & for better verbatim
\usepackage{subfig} % make it possible to include more than one captioned figure/table in a single float
% These packages are all incorporated in the memoir class to one degree or another...

%%% HEADERS & FOOTERS
\usepackage{fancyhdr} % This should be set AFTER setting up the page geometry
```

Brief Article

The Author

January 28, 2013

1 First section

Your text goes here.

1.1 A subsection

More text.

Figure 3 Screenshot of TeXworks on Ubuntu 12.10.

TeXworks[32] is a dedicated TeX editor that is included in MiKTeX and TeX Live. It was developed with the idea that a simple interface is better than a cluttered one, and thus to make it easier for people in their early days with LaTeX to get to what they want to do: write their documents. TeXworks originally came about precisely because a math professor wanted his students to have a better initial experience with LaTeX.

You can install TeXworks with the package manager of your Linux distribution or choose it as an install option in the Windows or Mac installer.

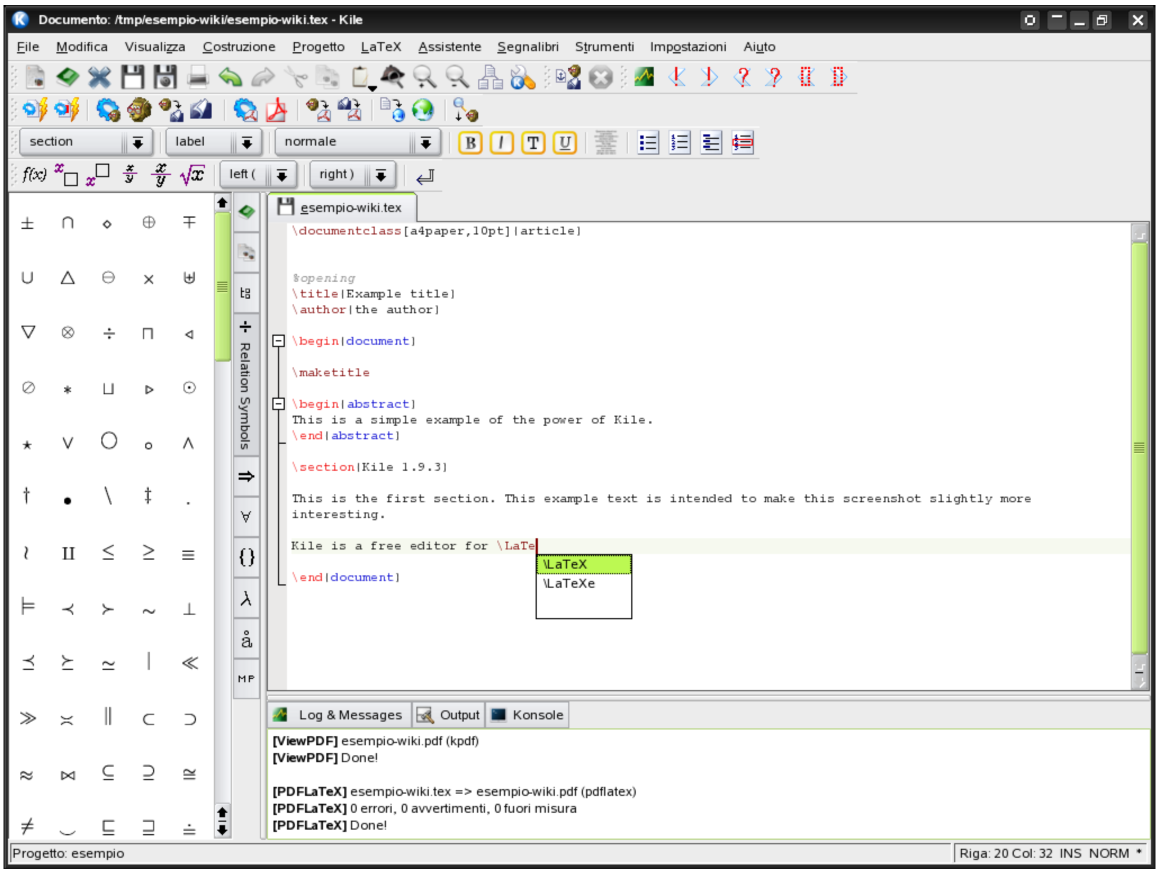
Vim

Vim[33] is another general purpose text editor for a wide variety of platforms including UNIX, Mac OS X and Windows. A variety of extensions exist including LaTeX Box[34] and Vim-LaTeX[35].

31 `http://texstudio.sourceforge.net/`
32 `http://en.wikipedia.org/wiki/TeXworks`
33 `http://en.wikipedia.org/wiki/Vim%20%28text%20editor%29`
34 `http://www.vim.org/scripts/script.php?script_id=3109`
35 `http://vim-latex.sourceforge.net/`

2.3.2. *BSD and GNU/Linux-only

Kile


Documento: /tmp/esempio-wiki/esempio-wiki.tex - Kile
File Modifica Visualizza Costruzione Progetto LaTeX Assistente Segnalibri Strumenti Impostazioni Aiuto
section
label
normale
left (
right)
Relation Symbols
esempio-wiki.tex
Log & Messages
Output
Konsole
Progetto: esempio
Riga: 20 Col: 32 INS NORM

```
\documentclass[a4paper,10pt]{article}


%opening
\title{Example title}
\author{the author}

\begin{document}

\maketitle

\begin{abstract}
This is a simple example of the power of Kile.
\end{abstract}

\section{Kile 1.9.3}

This is the first section. This example text is intended to make this screenshot slightly more
interesting.

Kile is a free editor for \LaTe
                                \LaTeX
                                \LaTeXe
\end{document}
```

```
[ViewPDF] esempio-wiki.pdf (kpdf)
[ViewPDF] Done!

[PDFLaTeX] esempio-wiki.tex => esempio-wiki.pdf (pdflatex)
[PDFLaTeX] 0 errori, 0 avvertimenti, 0 fuori misura
[PDFLaTeX] Done!
```

Figure 4 Screenshot of Kile[a].

a http://en.wikipedia.org/wiki/Kile

Kile[36] is a LaTeX editor for KDE[37] (cross platform), providing a powerful GUI for editing multiple documents and compiling them with many different TeX compilers. Kile is based on Kate editor, has a quick access toolbar for symbols, document structure viewer, a console and customizable build options. Kile can be run in all operating systems that can run KDE.

LaTeXila

LaTeXila[38] is another text editor for Linux (Gnome).

36 http://kile.sourceforge.net/
37 http://en.wikipedia.org/wiki/KDE_Software_Compilation_4
38 http://projects.gnome.org/latexila/

2.3.3. Mac OS X-only

TeXShop

TeXShop[39] is a TeXworks-like editor and previewer for Mac OS that is bundled with the MacTeX distribution. It uses multiple windows, one for editing the source, one for the preview, and one as a console for error messages. It offers one-click updating of the preview and allows easy crossfinding between the code and the preview by using CMD-click.

TeXnicle

TeXnicle[40] is a free editor for Mac OS that includes the ability to perform live updates. It includes a code library for the swift insertion of code and the ability to execute detailed word counts on documents. It also performs code highlighting and the editing window is customisable, permitting the user to select the font, colour, background colour of the editing environment. It is in active development.

Archimedes

Archimedes[41] is an easy-to-use LaTeX and Markdown editor designed from the ground up for Mac OS X. It includes a built-in LaTeX library, code completion support, live previews, macro support, integration with sharing services, and PDF and HTML export options. Archimedes's Magic Type feature lets users insert mathematical symbols just by drawing them on their MacBook's trackpad or Magic Trackpad.

2.3.4. Windows-only

LEd

LEd[42]

TeXnicCenter

TeXnicCenter[43] is a popular free and open source LaTeX editor for Windows. It also has a similar user interface to TeXmaker and Kile.

39 `http://www.uoregon.edu/~koch/texshop/`
40 `http://www.bobsoft-mac.de/texnicle/texnicle.html`
41 `http://www.mattrajca.com/archimedes`
42 `http://www.latexeditor.org/`
43 `http://www.texniccenter.org/`

WinEdt

WinEdt[44] is a powerful and versatile text editor with strong predisposition towards creation of LaTeX/TeX documents for Windows. It has been designed and configured to integrate with TeX Systems such as MiTeX or TeX Live. Its built-in macro helps in compiling the LaTeX source to the WYSIWYG-like DVI or PDF or PS and also in exporting the document to other mark-up languages as HTML or XML.

WinShell

WinShell[45]

2.3.5. Online solutions

To get started without needing to install anything, you can use a web-hosted service featuring a full TeX distribution and a web LaTeX editor.

- Authorea `https://authorea.com` is an integrated online framework for the creation of technical documents in collaboration. Authorea's frontend allows you to enter text in LaTeX or Markdown, as well as figures, and equations (in LaTeX or MathML). Authorea's versioning control system is entirely based on Git (every article is a Git repository).
- Google Documents[46] or LaTeX Lab[47] allows real-time simultaneous collaborative editing of text files for anyone with a Google account (and its option to make the document available through a URL makes local download and compilation easily scriptable).
- LIMSUP[48] is an online LaTeX editor allowing real time collaboration of LaTeX documents (announcement[49])
- Monkey TeX[50] is free and allows team sharing.
- Overleaf[51] is a secure, easy to use online LaTeX editor with integrated rapid preview - like EtherPad[52] for LaTeX. Start writing with one click (no signup required) and share the link. It supports real time preview, figures, bibliographies and custom styles.
- publications.li[53] is a real-time collaborative LaTeX editor running the open-source editor \BlueLatex[54].

44 `http://www.winedt.com/`
45 `http://www.winshell.de/`
46 `http://docs.google.com`
47 `http://docs.latexlab.org`
48 `http://www.limsup.com`
49 `http://www.digmi.org/2012/05/19/limsup-real-time-latex-collaboration/`
50 `http://monkeytex.bradcater.webfactional.com`
51 `https://www.overleaf.com`
52 `http://en.wikipedia.org/wiki/Etherpad`
53 `http://blue.publications.li`
54 `https://github.com/gnieh/bluelatex`

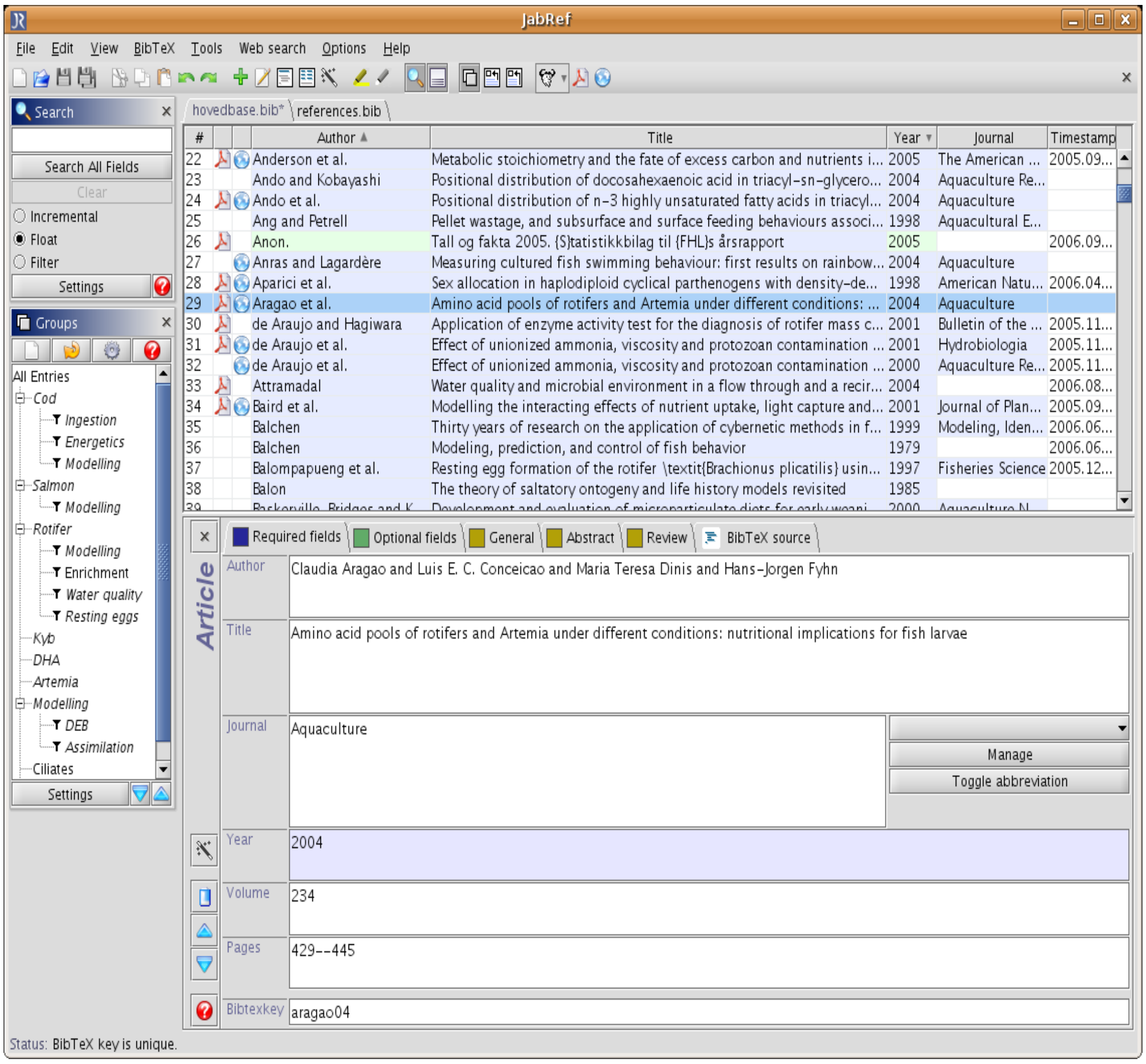
- ScribTeX.com[55] is one of the most mature systems available, with git push and pull access it allows for powerful version control. The new sign ups are now directed to use ShareLatex.com[56] however accounts are still available upon request.
- ShareLaTeX.com[57] is a secure cloud-based LaTeX editor offering unlimited free project. Premium accounts are available for extra features such as version control and Dropbox integration.
- SpanDeX[58] is a cloud-based LaTeX collaboration platform designed to make collaborating with LaTeX seamless and to reduce the learning curve to LaTeX. It offers simultaneous real-time editing and collaboration, live document preview, Dropbox integration, and a built-in LaTeX resource system.
- Verbosus[59] is a professional Online LaTeX Editor that supports collaboration with other users and is free to use. Merge conflicts can easily resolved by using a built-in merge tool that uses an implementation of the diff-algorithm to generate information required for a successful merge.

2.4. Bibliography management

Bibliography files (`*.bib`) are most easily edited and modified using a management system. These graphical user interfaces all feature a database form, where information is entered for each reference item, and the resulting text file can be used directly by BibTeX.

55 `http://www.scribtex.com/`
56 `https://www.sharelatex.com`
57 `https://www.sharelatex.com`
58 `http://spandex.io`
59 `http://www.verbosus.com`

2.4.1. Cross-platform



#	Author	Title	Year	Journal	Timestamp
22	Anderson et al.	Metabolic stoichiometry and the fate of excess carbon and nutrients i...	2005	The American ...	2005.09...
23	Ando and Kobayashi	Positional distribution of docosahexaenoic acid in triacyl-sn-glycero...	2004	Aquaculture Re...	
24	Ando et al.	Positional distribution of n-3 highly unsaturated fatty acids in triacyl...	2004	Aquaculture	
25	Ang and Petrell	Pellet wastage, and subsurface and surface feeding behaviours associ...	1998	Aquacultural E...	
26	Anon.	Tall og fakta 2005. {S}tatistikkbilag til {FHL}s årsrapport	2005		2006.09...
27	Anras and Lagardère	Measuring cultured fish swimming behaviour: first results on rainbow...	2004	Aquaculture	
28	Aparici et al.	Sex allocation in haplodiploid cyclical parthenogens with density-de...	1998	American Natu...	2006.04...
29	Aragao et al.	Amino acid pools of rotifers and Artemia under different conditions: ...	2004	Aquaculture	
30	de Araujo and Hagiwara	Application of enzyme activity test for the diagnosis of rotifer mass c...	2001	Bulletin of the ...	2005.11...
31	de Araujo et al.	Effect of unionized ammonia, viscosity and protozoan contamination ...	2001	Hydrobiologia	2005.11...
32	de Araujo et al.	Effect of unionized ammonia, viscosity and protozoan contamination ...	2000	Aquaculture Re...	2005.11...
33	Attramadal	Water quality and microbial environment in a flow through and a recir...	2004		2006.08...
34	Baird et al.	Modelling the interacting effects of nutrient uptake, light capture and...	2001	Journal of Plan...	2005.09...
35	Balchen	Thirty years of research on the application of cybernetic methods in f...	1999	Modeling, Iden...	2006.06...
36	Balchen	Modeling, prediction, and control of fish behavior	1979		2006.06...
37	Balompapueng et al.	Resting egg formation of the rotifer \textit{Brachionus plicatilis} usin...	1997	Fisheries Science	2005.12...
38	Balon	The theory of saltatory ontogeny and life history models revisited	1985		

Author: Claudia Aragao and Luis E. C. Conceicao and Maria Teresa Dinis and Hans-Jorgen Fyhn

Title: Amino acid pools of rotifers and Artemia under different conditions: nutritional implications for fish larvae

Journal: Aquaculture

Year: 2004

Volume: 234

Pages: 429--445

Bibtexkey: aragao04

Status: BibTeX key is unique.

Figure 5 Screenshot of JabRef[a].

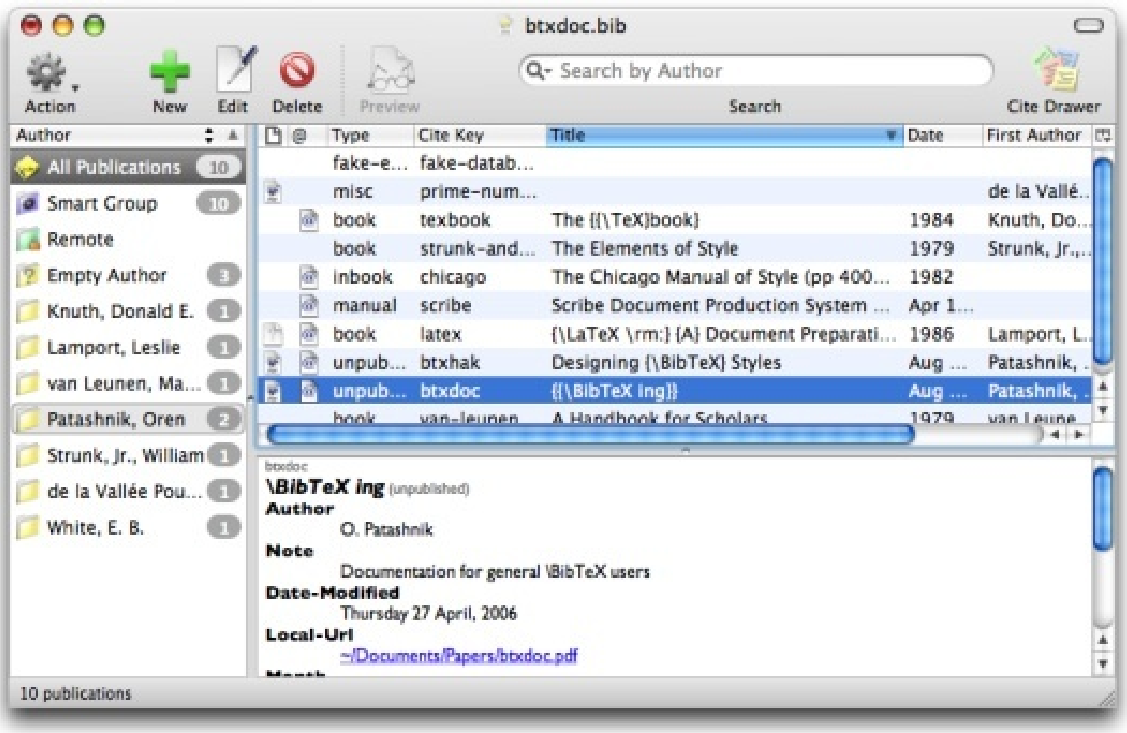
a http://en.wikipedia.org/wiki/JabRef

- JabRef[60]
- Mendeley[61]

60 http://jabref.sourceforge.net/

61 http://www.mendeley.com//

2.4.2. Mac OS X-only



Type	Cite Key	Title	Date	First Author
fake-e...	fake-datab...			
misc	prime-num...			de la Vallé...
book	texbook	The {{\TeX}book}	1984	Knuth, Do...
book	strunk-and...	The Elements of Style	1979	Strunk, Jr.,...
inbook	chicago	The Chicago Manual of Style (pp 400...	1982	
manual	scribe	Scribe Document Production System ...	Apr 1...	
book	latex	{\LaTeX \rm:} {A} Document Preparati...	1986	Lamport, L...
unpub...	btxhak	Designing {\BibTeX} Styles	Aug ...	Patashnik, ...
unpub...	btxdoc	{{\BibTeX ing}}	Aug ...	Patashnik, ...
book	van-leunen	A Handbook for Scholars	1979	van Leune...

btxdoc
\BibTeX ing (unpublished)
Author
O. Patashnik
Note
Documentation for general \BibTeX users
Date-Modified
Thursday 27 April, 2006
Local-Url
~/Documents/Papers/btxdoc.pdf

10 publications

Figure 6 Screenshot of BibDesk

- BibDesk[62] is a bibliography manager based on a BibTeX file. It imports references from the internet and makes it easy to organize references using tags and categories[63].

2.5. Viewers

Finally, you will need a viewer for the files LaTeX outputs. Normally LaTeX saves the final document as a `.dvi` (Device independent file format), but you will rarely want it to. DVI files do not contain embedded fonts and many document viewers are unable to open them.

Usually you will use a LaTeX compiler like `pdflatex` to produce a PDF file directly, or a tool like `dvi2pdf` to convert the DVI file to PDF format. Then you can view the result with any PDF viewer.

Practically all LaTeX distributions have a DVI viewer for viewing the default output of `latex` , and also tools such as `dvi2pdf` for converting the result automatically to PDF and PS formats.

Here follows a list of various PDF viewers.

62 `http://en.wikipedia.org/wiki/BibDesk`
63 BibDesk ^{`http://bibdesk.sourceforge.net/`}

- PDF.js (built-in modern browsers)
- Evince[64] (Linux GNOME)
- Foxit[65] (Windows)
- Okular[66] (Linux KDE)
- Preview (built-in Mac OS X)
- Skim[67] (Mac OS X)
- Sumatra PDF[68] (Windows)
- Xpdf[69] (Linux)
- Zathura[70] (Linux)

2.6. Tables and graphics tools

LaTeX is a document preparation system, it does not aim at being a spreadsheet tool nor a vector graphics tool.

If LaTeX can render beautiful tables in a dynamic and flexible manner, it will not handle the handy features you could get with a spreadsheet like dynamic cells and calculus. Other tools are better at that. The ideal solution is to combine the strength of both tools: build your dynamic table with a spreadsheet, and export it to LaTeX to get a beautiful table seamlessly integrated to your document. See Tables[71] for more details.

The graphics topic is a bit different since it is possible to write procedural graphics[72] from within your LaTeX document. Procedural graphics produce state-of-the-art results that integrates perfectly to LaTeX (*e.g.* no font change), but have a steep learning curve and require a lot of time to draw.

For easier and quicker drawings, you may want to use a WYSIWYG tool and export the result to a vector format like PDF. The drawback is that it will contrast in style with the rest of your document (font, size, etc.). Some tools have the capability to export to LaTeX, which will partially solve this issue. See Importing Graphics[73] for more details.

2.7. References

de:LaTeX/_Installation[74]

64 `https://wiki.gnome.org/Apps/Evince`
65 `http://www.foxitsoftware.com/Secure_PDF_Reader/`
66 `https://okular.kde.org/`
67 `http://skim-app.sourceforge.net/`
68 `http://www.sumatrapdfreader.org/free-pdf-reader.html`
69 `http://www.foolabs.com/xpdf/about.html`
70 `https://pwmt.org/projects/zathura/`
71 Chapter 14 on page 153
72 Chapter 44 on page 517
73 Chapter 17 on page 211
74 `http://de.wikibooks.org/wiki/LaTeX%2F_Installation`

3. Installing Extra Packages

Add-on features for LaTeX are known as packages. Dozens of these are pre-installed with LaTeX and can be used in your documents immediately. They should all be stored in subdirectories of `texmf/tex/latex` named after each package. The directory name ”texmf” stands for “TEX and METAFONT”. To find out what other packages are available and what they do, you should use the CTAN search page[1] which includes a link to Graham Williams’ comprehensive package catalogue.

A package is a file or collection of files containing extra LaTeX commands and programming which add new styling features or modify those already existing. There are two main file types: class files with `.cls` extension, and style files with `.sty` extension. There may be ancillary files as well. When you try to typeset a document which requires a package which is not installed on your system, LaTeX will warn you with an error message that it is missing. You can download updates to packages you already have (both the ones that were installed along with your version of LaTeX as well as ones you added). There is no limit to the number of packages you can have installed on your computer (apart from disk space!), but there is a configurable limit to the number that can be used inside any one LaTeX document at the same time, although it depends on how big each package is. In practice there is no problem in having even a couple of dozen packages active.

Most LaTeX installations come with a large set of pre-installed style packages, so you can use the package manager of the TeX distribution or the one on your system to manage them. See the automatic installation. But many more are available on the net. The main place to look for style packages on the Internet is CTAN[2]. Once you have identified a package you need that is not in your distribution, use the indexes on any CTAN server to find the package you need and the directory where it can be downloaded from. See the manual installation.

3.1. Automatic installation

If on an operating system with a package manager or a portage tree, you can often find packages in repositories.

With MikTeX there is a package manager that allows you to pick the package you want individually. As a convenient feature, upon the compilation of a file requiring non-installed packages, MikTeX will automatically prompt to install the missing ones.

With TeX Live, it is common to have the distribution packed into a few big packages. For example, to install something related to internationalization, you might have to install a

1 `http://www.ctan.org/search.html`
2 `http://www.ctan.org/`

package like `texlive-lang` . With TeX Live manually installed, use `tlmgr` to manage packages individually.

```
tlmgr install <package1> <package2> ...
tlmgr remove <package1> <package2> ...
```

The use of `tlmgr` is covered in the Installation[3] chapter.

If you cannot find the wanted package with any of the previous methods, see the manual installation.

3.2. Manual installation

3.2.1. Downloading packages

What you need to look for is usually two files, one ending in `.dtx` and the other in `.ins` . The first is a DOCTeX file, which combines the package program and its documentation in a single file. The second is the installation routine (much smaller). You must always download both files. If the two files are not there, it means one of two things:

- *Either* the package is part of a much larger bundle which you shouldn't normally update unless you change
  UNKNOWN TEMPLATE FULLBOOKNAME
  version of LaTeX;
- *or* it's an older or relatively simple package written by an author who did not use a `.dtx` file.

Download the package files to a temporary directory. There will often be a `readme.txt` with a brief description of the package. You should of course read this file first.

3.2.2. Installing a package

There are five steps to installing a LaTeX package. (These steps can also be used on the pieces of a complicated package you wrote yourself; in this case, skip straight to Step 3.)

1. **Extract the files** Run LaTeX on the `.ins` file. That is, open the file in your editor and process it as if it were a LaTeX document (which it is), or if you prefer, type latex followed by the `.ins` filename in a command window in your temporary directory. This will extract all the files needed from the `.dtx` file (which is why you must have both of them present in the temporary directory). Note down or print the names of the files created if there are a lot of them (read the log file if you want to see their names again).

2. **Create the documentation** Run LaTeX on the `.dtx` file. You might need to run it twice or more, to get the cross-references right (just like any other LaTeX document). This will create a `.dvi` file of documentation explaining what the package is for and how

3 Chapter 2 on page 11

to use it. If you prefer to create PDF then run pdfLaTeX instead. If you created a `.idx` as well, it means that the document contains an index, too. If you want the index to be created properly, follow the steps in the indexing[4] section. Sometimes you will see that a `.glo` (glossary) file has been produced. Run the following command instead:

```
makeindex -s gglo.ist -o name.gls name.glo
```

3. **Install the files** While the documentation is printing, move or copy the package files from your temporary directory to the right place[s] in your TeX local installation directory tree. Packages installed by hand should always be placed in your "local" directory tree, *not* in the directory tree containing all the pre-installed packages. This is done to *a)* prevent your new package accidentally overwriting files in the main TeX directories; and *b)* avoid your newly-installed files being overwritten when you next update your version of TeX.

For a TDS(TeX Directory Structure)-conformant system, your "local installation directory tree" is a folder and its subfolders. The outermost folder should probably be called `texmf-local/` or `texmf/` . Its location depends on your system:

- MacTeX: `Users/`*`username`*` /Library/texmf/` .
- Unix-type systems: Usually `~/texmf/` .
- MikTeX: Your local directory tree can be any folder you like, as long as you then register it as a user-managed texmf directory (see `http://docs.miktex.org/manual/localadditions.html#id573803)`

The "right place" sometimes causes confusion, especially if your TeX installation is old or does not conform to the TeX Directory Structure(TDS). For a TDS-conformant system, the "right place" for a LaTeX `.sty` file is a suitably-named subdirectory of `texmf/tex/latex/` . "Suitably-named" means sensible and meaningful (and probably short). For a package like paralist, for example, I'd call the directory `texmf/tex/latex/paralist` .

Often there is just a `.sty` file to move, but in the case of complex packages there may be more, and they may belong in different locations. For example, new BibTeX packages or font packages will typically have several files to install. This is why it is a good idea to create a sub-directory for the package rather than dump the files into misc along with other unrelated stuff. If there are configuration or other files, read the documentation to find out if there is a special or preferred location to move them to.

Where to put files from packages		
Type	**Directory (under `texmf/` or `texmf-local/`)**	**Description**
.afm	fonts/afm/*foundry* /*typeface*	Adobe Font Metrics for Type 1 fonts
.bst	bibtex/bst/*packagename*	BibTeX style
.cls	tex/latex/base	Document class file
.dvi	doc	package documentation
.enc	fonts/enc	Font encoding

4 Chapter 34 on page 421

Where to put files from packages		
Type	**Directory (under `texmf/` or `texmf-local/`)**	**Description**
.fd	tex/latex/mfnfss	Font Definition files for METAFONT fonts
.fd	tex/latex/psnfss	Font Definition files for PostScript Type 1 fonts
.map	fonts/map/	Font mapping files
.mf	fonts/source/public/*typeface*	METAFONT outline
.pdf	doc	package documentation
.pfb	fonts/type1/*foundry* /*typeface*	PostScript Type 1 outline
.sty	tex/latex/*packagename*	Style file: the normal package content
.tex	doc	TeX source for package documentation
.tex	tex/plain/*packagename*	Plain TeX macro files
.tfm	fonts/tfm/*foundry* /*typeface*	TeX Font Metrics for METAFONT and Type 1 fonts
.ttf	fonts/truetype/*foundry* /*typeface*	TrueType font
.vf	fonts/vf/*foundry* /*typeface*	TeX virtual fonts
others	tex/latex/*packagename*	other types of file unless instructed otherwise

For most fonts on CTAN, the *foundry* is `public` .

4. **Update your index** Finally, run your TeX indexer program to update the package database. This program comes with every modern version of TeX and has various names depending on the LaTeX distribution you use. (Read the documentation that came with your installation to find out which it is, or consult `http://www.tug.org/fonts/fontinstall.html#fndb`):

- teTeX, TeX Live, fpTeX: `texhash`
- web2c: `mktexlsr`
- MacTeX: MacTeX appears to do this for you.
- MikTeX: `initexmf --update-fndb` (or use the GUI)
- MiKTeX 2.7 or later versions, installed on Windows XP through Windows 7: Start -> All Programs -> MikTex -> Settings. In Windows 8 use the keyword *Settings* and choose the option of *Settings* with the MiKTex logo. In *Settings* menu choose the first tab and click on *Refresh FNDB* -button (MikTex will then check the Program Files directory and update the list of File Name DataBase). After that just verify by clicking 'OK'.

⚠ **Warning**

This step is utterly essential, otherwise nothing will work.

5. **Update font maps** If your package installed any TrueType or Type 1 fonts, you need to update the font mapping files *in addition* to updating the index. Your package author

should have included a `.map` file for the fonts. The map updating program is usually some variant on `updmap` , depending on your distribution:

- TeX Live and MacTeX: `updmap --enable Map=`*`mapfile`*` .map` (if you installed the files in a personal tree) or `updmap-sys --enable Map=`*`mapfile`*` .map` (if you installed the files in a system directory).
- MikTeX: Run `initexmf --edit-config-file updmap` , add the line "`Map `*`mapfile`*` .map` to the file that opens, then run `initexmf --mkmaps` .

See `http://www.tug.org/fonts/fontinstall.html.`

The reason this process has not been automated widely is that there are still thousands of installations which do not conform to the TDS, such as old shared Unix systems and some Microsoft Windows systems, so there is no way for an installation program to guess where to put the files: you have to know this. There are also systems where the owner, user, or installer has chosen not to follow the recommended TDS directory structure, or is unable to do so for political or security reasons (such as a shared system where the user cannot write to a protected directory). The reason for having the `texmf-local` directory (called `texmf.local` on some systems) is to provide a place for local modifications or personal updates, especially if you are a user on a shared or managed system (Unix, Linux, VMS, Windows NT/2000/XP, etc.) where you may not have write-access to the main TeX installation directory tree. You can also have a personal `texmf` subdirectory in your own login directory. Your installation must be configured to look in these directories first, however, so that any updates to standard packages will be found there before the superseded copies in the main `texmf` tree. All modern TeX installations should do this anyway, but if not, you can edit `texmf/web2c/texmf.cnf` yourself.

3.3. Checking package status

The universal way to check if a file is available to TeX compilers is the command-line tool `kpsewhich` .

```
$ kpsewhich tikz
/usr/local/texlive/2012/texmf-dist/tex/plain/pgf/frontendlayer/tikz.tex
```

`kpsewhich` will actually search for files only, not for packages. It returns the path to the file. For more details on a specific package use the command-line tool `tlmgr` (TeX Live only):

```
tlmgr info <package>
```

The `tlmgr` tool has lot more options. To consult the documentation:

```
tlmgr help
```

3.4. Package documentation

To find out what commands a package provides (and thus how to use it), you need to read the documentation. In the `texmf/doc` subdirectory of your installation there should be directories full of .dvi files, one for every package installed. This location is distribution-specific, but is *typically* found in:

Distribution	**Path**
MacTeX	`/Library/TeX/Documentation/texmf-doc/latex`
MiKTeX	`%MIKTEX_DIR%\doc\latex`
TeX Live	`$TEXMFDIST/doc/latex`

Generally, *most* of the packages are in the `latex` subdirectory, although other packages (such as BibTeX and font packages) are found in other subdirectories in `doc` . The documentation directories have the same name of the package (e.g. `amsmath`), which generally have one or more relevant documents in a variety of formats (`dvi` , `txt` , `pdf` , etc.). The documents generally have the same name as the package, but there are exceptions (for example, the documentation for `amsmath` is found at `latex/amsmath/amsdoc.dvi`). If your installation procedure has not installed the documentation, the DVI files can all be downloaded from CTAN. Before using a package, you should read the documentation carefully, especially the subsection usually called ”User Interface”, which describes the commands the package makes available. You cannot just guess and hope it will work: you have to read it and find out.

You can usually automatically open any installed package documentation with the *texdoc* command:

```
texdoc <package-name>
```

3.5. External resources

The best way to look for LaTeX packages is the already mentioned CTAN: Search[5]. Additional resources form The TeX Catalogue Online[6]:

- Alphabetic catalogue[7]
- With brief descriptions[8]
- Topical catalogue[9] with packages sorted systematically
- Hierarchical[10] mirroring the CTAN folder hierarchy

5 `http://tug.ctan.org/search.html`
6 `http://www.ctan.org/tex-archive/help/Catalogue/catalogue.html`
7 `http://www.ctan.org/tex-archive/help/Catalogue/alpha.html`
8 `http://www.ctan.org/tex-archive/help/Catalogue/brief.html`
9 `http://www.ctan.org/tex-archive/help/Catalogue/bytopic.html`
10 `http://www.ctan.org/tex-archive/help/Catalogue/hier.html`

3.6. See Also

- LaTeX/Package Reference[11]

11 Chapter 62 on page 659

4. Basics

This tutorial is aimed at getting familiar with the bare bones of LaTeX[1].

Before starting, ensure you have LaTeX installed on your computer (see Installation[2] for instructions of what you will need).

- We will first have a look at the LaTeX syntax.
- We will create our first LaTeX document.
- Then we will take you through how to feed this file through the LaTeX system to produce quality output, such as postscript or PDF.
- Finally we will have a look at the file names and types.

4.1. The LaTeX syntax

LaTeX uses a markup language in order to describe document structure and presentation. LaTeX converts your source text, combined with the markup, into a high quality document. For the purpose of analogy, web pages work in a similar way: the HTML is used to describe the document, but it is your browser that presents it in its full glory - with different colours, fonts, sizes, etc.

The input for LaTeX is a plain text[3] file. You can create it with any text editor. It contains the text of the document, as well as the commands that tell LaTeX how to typeset the text.

A minimal example looks something like the following (the commands will be explained later):

```
\documentclass{article}

\begin{document}
Hello world!
\end{document}
```

4.1.1. Spaces

The LaTeX compiler normalises whitespace so that whitespace characters, such as [space] or [tab], are treated uniformly as ”space”: several consecutive ”spaces” are treated as one, ”space” opening a line is generally ignored, and a single line break also yields “space”. A double line break (an empty line), however, defines the end of a paragraph; multiple empty

1 Chapter 1 on page 5
2 Chapter 2 on page 11
3 http://en.wikipedia.org/wiki/plain%20text

lines are also treated as the end of a paragraph. An example of applying these rules is presented below: the left-hand side shows the user's input (.tex), while the right-hand side depicts the rendered output (.dvi/.pdf/.ps).

```
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.

4.1.2. Reserved Characters

The following symbols are reserved characters that either have a special meaning under LaTeX or are unavailable in all the fonts. If you enter them directly in your text, they will normally not print but rather make LaTeX 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:

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

The backslash character \ *cannot* be entered by adding another backslash in front of it (\\); this sequence is used for line breaking. For introducing a backslash in math mode, you can use `\backslash` instead.

The commands \~ and \^ produce respectively a tilde and a hat which is placed over the next letter. For example `\~n` gives ñ. That's why you need braces to specify there is no letter as argument. You can also use `\textasciitilde` and `\textasciicircum` to enter these characters; or other commands [4].

If you want to insert text that might contain several particular symbols (such as URIs), you can consider using the `\verb` command, which will be discussed later in the section on formatting[5]. For source code, see Source Code Listings[6]

The 'less than' (<) and 'greater than' (>) characters are the only visible ASCII characters (not reserved) that will not print correctly. See Special Characters[7] for an explanation and a workaround.

4 `http://tex.stackexchange.com/questions/9363/how-does-one-insert-a-backslash-or-a-tilde-into-latex`
5 `http://en.wikibooks.org/wiki/LaTeX%2FFormatting`
6 Chapter 32 on page 393
7 Chapter 11.3 on page 127

Non-ASCII characters (*e.g.* accents, diacritics) can be typed in directly for most cases. However you must *configure* the document appropriately. The other symbols and many more can be printed with special commands in mathematical formulae or as accents. We will tackle this issue in Special Characters[8].

4.1.3. LaTeX groups

A group is basically defined by a pair of braces. The range of commands put between braces is limited to them. The `\begingroup` and `\endgroup` commands are equivalent to opening brace and closing brace.

Example:

```
{
\bf This is bold.
}
This is no longer bold.
```

For some commands it is important to restrict their range of action, and that's where groups come to be very useful.

4.1.4. LaTeX environments

Environments in LaTeX have a role that is quite similar to commands, but they usually have effect on a wider part of the document. Their syntax is:

```
\begin{environmentname}
text to be influenced
\end{environmentname}
```

Between the `\begin` and the `\end` you can put other commands and nested environments. The internal mechanism of environments defines a group, which makes its usage safe (no influence on the other parts of the document). In general, environments can accept arguments as well, but this feature is not commonly used and so it will be discussed in more advanced parts of the document.

Anything in LaTeX can be expressed in terms of commands and environments.

4.1.5. 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 non-letter.

Some commands need an argument, 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 general syntax is:

8 Chapter 11 on page 123

```
\commandname[option1,option2,...]{argument1}{argument2}...
```

Most standard LaTeX commands have a *switch* equivalent. Switches have no arguments but apply on the rest of the scope, *i.e.* the current group or environment. A switch should (almost) never be called outside of any scope, otherwise it will apply on the rest of the document.

⚠ **Warning**

Commands with arguments and switches should not be confused. This is a very common error!

Example:

```
% \emph is a command with argument, \em is a switch.
\emph{emphasized text}, this part is normal % Correct
{\em emphasized text}, this part is normal % Correct

\em emphasized text, this part is normal % Incorrect
\em{emphasized text}, this part is normal % Incorrect
```

4.1.6. Comments

When LaTeX encounters a % character while processing an input file, it ignores the rest of the current line, the line break, 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

Note that the % character can be used to split long input lines that do not allow whitespace or line breaks, as with Supercalifragilisticexpialidocious above.

The core LaTeX language does not have a predefined syntax for commenting out regions spanning multiple lines. Refer to multiline comments[9] for simple workarounds.

9 Chapter 7.6.2 on page 85

4.2. Our first document

Now we can create our first document. We will produce the absolute bare minimum that is needed in order to get some output; the well known **Hello World!** approach will be suitable here.

- Open your favorite text-editor. vim[10], emacs[11], Notepad++, and other text editors will have syntax highlighting that will help to write your files.
- Reproduce the following text in your editor. This is the LaTeX source.

```
% hello.tex - Our first LaTeX example!
\documentclass{article}
\begin{document}
Hello World!
\end{document}
```

- Save your file as `hello.tex` .

When picking a name for your file, make sure it bears a `.tex` extension.

4.2.1. What does it all mean?

`% hello.tex - Our first LaTeX example!`	The first line is a *comment* . This is because it begins with the percent symbol (%); when LaTeX sees this, it simply ignores the rest of the line. Comments are useful for people to annotate parts of the source file. For example, you could put information about the author and the date, or whatever you wish.
`\documentclass{article}`	This line is a command and tells LaTeX to use the `article` document class. A document class file defines the formatting, which in this case is a generic article format. The handy thing is that if you want to change the appearance of your document, substitute article for another class file that exists.
`\begin{document}`	This line is the beginning of the environment called `document`; it alerts LaTeX that content of the document is about to commence. Anything above this command is known generally to belong in the *preamble* .
`Hello World!`	This was the only actual line containing real content - the text that we wanted displayed on the page.
`\end{document}`	The `document` environment ends here. It tells LaTeX that the document source is complete, anything after this line will be ignored.

As we have said before, each of the LaTeX commands begins with a backslash (\). This is LaTeX's way of knowing that whenever it sees a backslash, to expect some commands. Comments are not classed as a command, since all they tell LaTeX is to ignore the line. Comments never affect the output of the document.

10 `http://en.wikibooks.org/wiki/Learning%20the%20vi%20Editor%2FVim`
11 `http://en.wikibooks.org/wiki/emacs`

4.3. Compilation

4.3.1. Compilation process

The general concept is to transform a plain text document into a publishable format, mosty a DVI, PS or PDF file. This process is called *compilation* , which is done by an executable file called a *compiler* .

There are two main compilers.

- `tex` compiler reads a TeX `.tex` file and creates a `.dvi` .
- `pdftex` compiler reads a TeX `.tex` file and creates a `.pdf` .

These compilers are basically used to compile Plain TeX, not LaTeX. There is no such LaTeX compiler since LaTeX is just a bunch of macros for TeX. However, there are two executables related to the previous compilers:

- `latex` executable calls `tex` with LaTeX initialization files, reads a LaTeX `.tex` file and creates a `.dvi`
- `pdflatex` executable calls `pdftex` with LaTeX initialization files, reads a LaTeX `.tex` file and creates a `.pdf`

If you compile a Plain TeX document with a LaTeX compiler (such as `pdflatex`) it will work while the opposite is not true: if you try to compile a LaTeX source with a TeX compiler you will get many errors.

As a matter of fact, following your operating system `latex` and `pdflatex` are simple scripts or symbolic links.

Most of the programs should be already within your LaTeX distribution; the others come with Ghostscript[12], which is a free and multi-platform software as well. Here are common programs you expect to find in any LaTeX distribution:

- `dvi2ps` converts the `.dvi` file to `.ps` (postscript).
- `dvi2pdf` converts the `.dvi` file to `.pdf` (`dvi2pdfm` is an improved version).

and with Ghostscript:

- `ps2pdf` and `pdf2ps` converts the `.ps` file to `.pdf` and vice-versa.

When LaTeX was created, the only format it could create was DVI; later PDF support was added by `pdflatex` . PDF files can be created with both `pdflatex` and `dvipdfm` . The output of `pdflatex` takes direct advantage of modern features of PDF such as hyperlinks and embedded fonts, which are not part of DVI. Passing through DVI imposes limitations of its older format. On the other hand, some packages, such as PSTricks, exploit the process of conversion to DVI, and therefore will not work with pdflatex. Some of those packages embed information in the DVI that doesn't appear when the DVI is viewed, but reemerges when the DVI is converted to another, newer format.

You would write your document slightly differently depending on the compiler you are using (`latex` or `pdflatex`). But as we will see later it is possible to add a sort of abstraction

12 `http://en.wikibooks.org/wiki/Ghostscript`

layer to hide the details of which compiler you're using, while the compiler can handle the translation itself.

The following diagram shows the relationships between the LaTeX source code and the formats you can create from it:

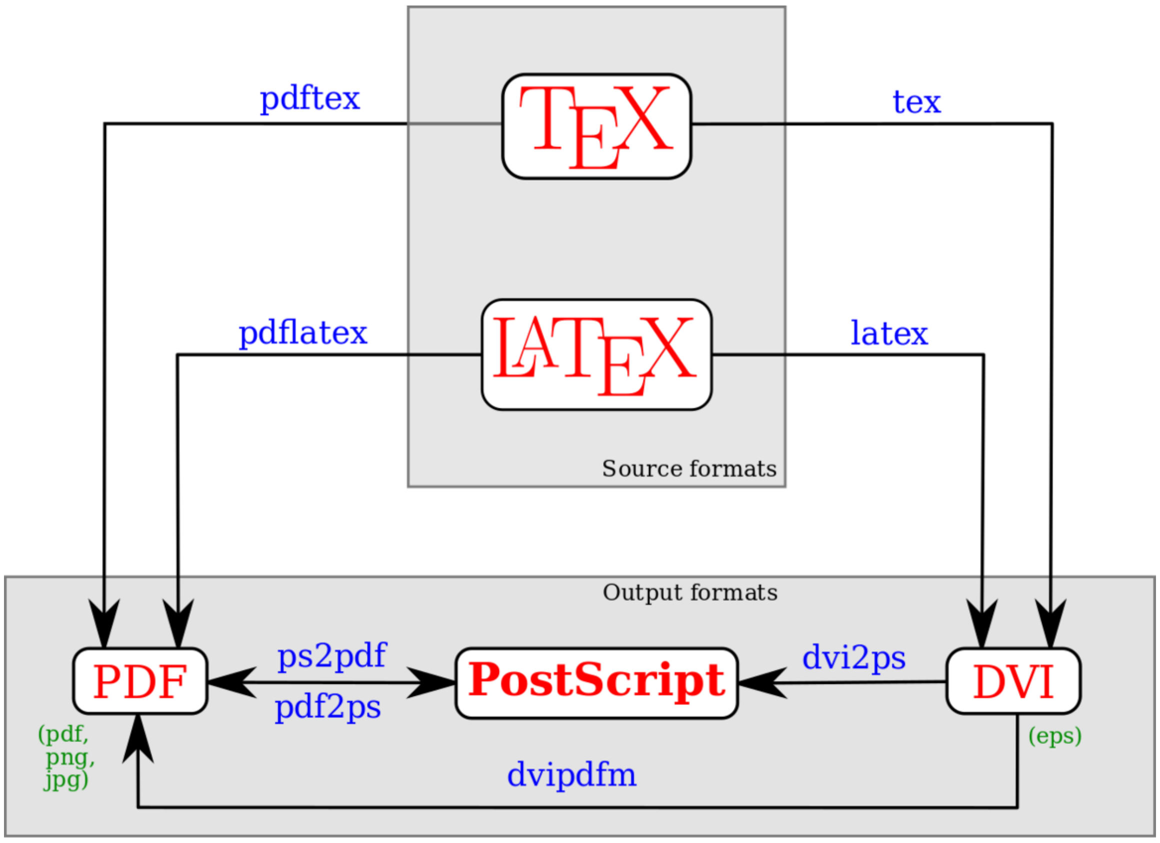


Figure 7

The boxed red text represents the file formats, the blue text on the arrows represents the commands you have to use, the small dark green text under the boxes represents the image formats that are supported. Any time you pass through an arrow you lose some information, which might decrease the features of your document. Therefore, you should choose the shortest route to reach your target format. This is probably the most convenient way to obtain an output in your desired format anyway. Starting from a LaTeX source, the best way is to use only *latex* for a DVI output or *pdflatex* for a PDF output, converting to PostScript only when it is necessary to print the document.

Chapter ../Export To Other Formats/[13] discusses more about exporting LaTeX source to other file formats.

13 Chapter 57 on page 627

4.3.2. Generating the document

It is clearly not going to be the most exciting document you have ever seen, but we want to see it nonetheless. I am assuming that you are at a command prompt, already in the directory where `hello.tex` is stored. LaTeX itself does not have a GUI (graphical user interface), since it is just a program that crunches away at your input files, and produces either a DVI or PDF file. Some LaTeX installations feature a graphical front-end where you can click LaTeX into compiling your input file. On other systems there might be some typing involved, so here is how to coax LaTeX into compiling your input file on a text based system. Please note: this description assumes that you already have a working LaTeX installation on your computer.

1. Type the command: `latex hello` (the `.tex` extension is not required, although you can include it if you wish)
2. Various bits of info about LaTeX and its progress will be displayed. If all went well, the last two lines displayed in the console will be:

```
Output written on hello.dvi (1 page, 232 bytes).
Transcript written on hello.log.
```

This means that your source file has been processed and the resulting document is called *hello.dvi* , which takes up 1 page and 232 bytes of space. Now you may view the DVI file. On Unix with X11 you can type `xdvi foo.dvi` , on Windows you can use a program called *yap* (yet another previewer). (Now evince and okular, the standard document viewers for many Linux distributions are able to view DVI files.)

This way you created the DVI file, but with the same source file you can create a PDF document. The steps are exactly the same as before, but you have to replace the command `latex` with `pdflatex` :

1. Type the command: `pdflatex hello` (as before, the `.tex` extension is not required)
2. Various bits of info about LaTeX and its progress will be displayed. If all went well, the last two lines displayed in the console will be:

```
Output written on hello.pdf (1 page, 5548 bytes).
Transcript written on hello.log.
```

you can notice that the PDF document is bigger than the DVI, even if it contains exactly the same information. The main differences between the DVI and PDF formats are:

- **DVI** needs less disk space and it is faster to create. It does not include the fonts within the document, so if you want the document to be viewed properly on another computer, there must be all the necessary fonts installed. It does not support any interactivity such as hyperlinks or animated images. DVI viewers are not very common, so you can consider using it for previewing your document while typesetting.
- **PDF** needs more disk space and it is slower to create, but it includes all the necessary fonts within the document, so you will not have any problem of portability. It supports internal and external hyperlinks. It also supports advanced typographic features: hanging punctuation[14], font expansion and margin kerning resulting in more flexibility available

14 `http://en.wikipedia.org/wiki/Hanging%20punctuation`

to the TeX engine and better looking output. Nowadays it is the *de facto* standard for sharing and publishing documents, so you can consider using it for the final version of your document.

About now, you saw you can create both DVI and PDF document from the same source. This is true, but it gets a bit more complicated if you want to introduce images or links. This will be explained in detail in the next chapters, but for now assume you can compile in both DVI and PDF without any problem.

Note, in this instance, due to the simplicity of the file, you only need to run the LaTeX command once. However, if you begin to create complex documents, including bibliographies and cross-references, etc, LaTeX needs to be executed multiple times to resolve the references. But this will be discussed in the future when it comes up.

4.3.3. Autobuild Systems

Compiling using only the `latex` binary can be quite tricky as soon as you start working on more complex documents as previously stated. A number of programs exist to automatically read in a TeX document and run the appropriate compilers the appropriate number of times. For example, `latexmk` can generate a PDF from most TeX files simply:

```
$ latexmk -pdf file.tex
```

Note that most editors[15] will take care of it for you.

4.3.4. Compressed PDF

For a PDF output, you may have noticed that the output PDF file is not always the same size depending on the engine you used to compile the file. So `latex → dvips → ps2pdf` will usually be much smaller than `pdflatex` . If you want `pdflatex` features along with a small output file size, you can use the Ghostscript command:

```
$ gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -sOutputFile="Compressed.pdf"
"Original.pdf"
```

4.4. Files

4.4.1. Picking suitable filenames

Never, ever use directories (folders) or file names that contain spaces. Although your operating system probably supports them, some don't, and they will only cause grief and tears with TeX. Make filenames as short or as long as you wish, but strictly avoid spaces. Stick

15 Chapter 2.3 on page 17

to lower-case letters without accents (a-z), the digits 0-9, the hyphen (−), and only one full point or period (.) to separate the file extension (somewhat similar to the conventions for a good Web URL): it will let you refer to TeX files over the Web more easily and make your files more portable. Some operating systems do not distinguish between upper-case and lower-case letters, others do. Therefore it's best not to mix them.

4.4.2. Ancillary files

The TeX compilers are single-pass processes. It means that there is no way for a compiler to *jump* around the document, which would be useful for the table of contents and references. Indeed the compiler cannot guess at which page a specific section is going to be printed, so when the table of contents is printed before the upcoming sections, it cannot set the page numbers.

To circumvent this issue, many LaTeX commands which need to *jump* use ancillary files which usually have the same file name as the current document but a different extension. It stores temporary data into these files and use them for the next compilation. So to have an up-to-date table of contents, you need to compile the document twice. There is no need to re-compile if no section moved.

For example, the temporary file for the table of contents data is `filename.toc` .

None of these files contains unrecoverable information. It means you can delete them safely, compiling will regenerate them automatically.

⚠ **Warning**

The only important file types are `.tex` , `.cls` and `.sty` , `.bib` and `.bst` for BibTeX, these are not temporary and should not be deleted.

When you work with various capabilities of LaTeX (index, glossaries, bibliographies, etc.) you will soon find yourself in a maze of files with various extensions and probably no clue. The following list explains the most common file types you might encounter when working with TeX:

Common file extensions in LaTeX	
`.aux`	A file that transports information from one compiler run to the next. Among other things, the `.aux` file is used to store information associated with cross-references.
`.bbl`	Bibliography file output by BiBTeX and used by LaTeX
`.bib`	Bibliography database file. (where you can store a list of full bibliographic citations)
`.blg`	BiBTeX log file. (errors are logged here)
`.bst`	BiBTeX style file.
`.cls`	Class files define what your document looks like. They are selected with the `\documentclass` command.

Common file extensions in LaTeX	
`.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`	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.
`.fd`	Font description file telling LaTeX about new fonts.
`.dvi`	Device Independent File. This is the main result of a LaTeX compile run with *latex* . 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.
`.pdf`	Portable Document Format. This is the main result of a LaTeX compile run with *pdflatex* . You can look at its content or print it with any PDF viewer.
`.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 contents.
`.lof`	This is like `.toc` but for the list of figures.
`.lot`	And again the same for the list of tables.
`.idx`	If your document contains an index. LaTeX stores all the words that go into the index in this file. Process this file with makeindex.
`.ind`	The processed .idx file, ready for inclusion into your document on the next compile cycle.
`.ilg`	Logfile telling what makeindex did.
`.sty`	LaTeX Macro package. This is a file you can load into your LaTeX document using the \usepackage command.
`.tex`	LaTeX or TeX input file. It can be compiled with latex.
`.out`	hyperref package file, just one for the master file.

4.5. And what now?

4.5.1. Common Elements

See Document Structure[16] and the **Common Elements** part for all the common features that belong to every type of document.

4.5.2. Non-English documents and special characters

LaTeX has some nice features for most languages in the world. You can tell LaTeX to follow typography rules of the target language, ease special characters input, and so on. See Special Characters[17] and Internationalization[18].

16 Chapter 5 on page 51
17 Chapter 11 on page 123
18 Chapter 12 on page 133

4.5.3. Modular document

See Modular Documents[19] for good recommendations about the way to organize big projects into multiple files.

4.5.4. Questions and Issues

We highly urge you to read the FAQ[20] if you have issues about basic features, or if you want to read essential recommendations. For the more specific questions and issues, refer to the Tips and Tricks[21] page. If you cannot find what you want here, use the Q&A[22] page.

4.5.5. Macros for the utmost efficiency

The full power of LaTeX resides in macros. They make your documents very dynamic and flexible. See the dedicated part[23].

4.5.6. Working in a team

See chapter ../Collaborative Writing of LaTeX Documents/[24].

19 Chapter 55 on page 607
20 Chapter 58 on page 637
21 Chapter 59 on page 643
22 `http://en.wikibooks.org/wiki/LaTeX%2FQ%26A`
23 Chapter 51 on page 569
24 Chapter 56 on page 615

Part II.

Common Elements

5. Document Structure

The main point of writing a text 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 semantic 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. LaTeX allows users to structure their documents with a variety of hierarchical constructs, including chapters, sections, subsections and paragraphs.

5.1. Global structure

When LaTeX processes an input file, it expects it to follow a certain structure. Thus every input file must contain the commands

```
\documentclass{...}

\begin{document}
...
\end{document}
```

The area between `\documentclass{...}` and `\begin{document}` is called the *preamble* . It normally contains commands that affect the entire document.

After the preamble, the text of your document is enclosed between two commands which identify the beginning and end of the actual document:

```
\begin{document}
...
\end{document}
```

You would put your text where the dots are. The reason for marking off the beginning of your text is that LaTeX allows you to insert extra setup specifications before it (where the blank line is in the example above: we'll be using this soon). The reason for marking off the end of your text is to provide a place for LaTeX to be programmed to do extra stuff automatically at the end of the document, like making an index.

A useful side-effect of marking the end of the document text is that you can store comments or temporary text underneath the `\end{document}` in the knowledge that LaTeX will never try to typeset them:

```
\end{document}
```

...

5.2. Preamble

5.2.1. Document classes

When processing an input file, LaTeX needs to know the type of document the author wants to create. This is specified with the `\documentclass` command. It is recommended to put this declaration at the very beginning.

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

Here, `class` specifies the type of document to be created. The LaTeX distribution provides additional classes for other documents, including letters and slides. It is also possible to create your own, as is often done by journal publishers, who simply provide you with their own class file, which tells LaTeX how to format your content. But we'll be happy with the standard article class for now. The `options` parameter customizes the behavior of the document class. The options have to be separated by commas.

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 11 points, and to produce a layout suitable for double sided printing on A4 paper.

Here are some document classes that can be used with LaTeX:

Document Classes	
`article`	For articles in scientific journals, presentations, short reports, program documentation, invitations, ...
`IEEEtran`	For articles with the IEEE Transactions format.
`proc`	A class for proceedings based on the article class.
`minimal`	Is as small as it can get. It only sets a page size and a base font. It is mainly used for debugging purposes.
`report`	For longer reports containing several chapters, small books, thesis, ...
`book`	For real books.
`slides`	For slides. The class uses big sans serif letters.
`memoir`	For changing sensibly the output of the document. It is based on the `book` class, but you can create any kind of document with it `http://www.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf`
`letter`	For writing letters.
`beamer`	For writing presentations (see LaTeX/Presentations[1]).

1 Chapter 41 on page 491

The standard document classes that are a part of LaTeX are built to be fairly generic, which is why they have a lot of options in common. Other classes may have different options (or none at all). Normally, third party classes come with some documentation to let you know. The most common options for the standard document classes are listed in the following table:

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`; However, many European distributions of TeX now come pre-set for A4, not Letter, and this is also true of all distributions of pdfLaTeX. Besides that, `a5paper, b5paper, executivepaper`, and `legalpaper` can be specified.
`fleqn`	Typesets displayed formulas left-aligned instead of centered.
`leqno`	Places the numbering of formulas 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.
`twocolumn`	Instructs LaTeX to typeset the document in two columns instead of one.
`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.
`landscape`	Changes the layout of the document to print in landscape mode.
`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.
`draft`	makes LaTeX indicate hyphenation and justification problems with a small square in the right-hand margin of the problem line so they can be located quickly by a human. It also suppresses the inclusion of images and shows only a frame where they would normally occur.

For example, if you want a report to be in 12pt type on A4, but printed one-sided in draft mode, you would use:

```
\documentclass[12pt,a4paper,oneside,draft]{report}
```

5.2.2. 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, colored text or source code from a file into your document, you need to enhance the capabilities of LaTeX. Such enhancements are called packages. Some packages come with the LaTeX base distribution. Others are provided separately. Modern TeX distributions come with a large number of packages pre-installed. Packages are activated with the

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

command, where package is the name of the package and options is a list of keywords that trigger special features in the package. For example, to use the `color` package, which lets you typeset in colors, you would type:

```
\documentclass[11pt,a4paper,oneside]{report}

\usepackage{color}

\begin{document}
...
\end{document}
```

You can include several package names in one `\usepackage` command by separating the names with commas, like this:

```
\usepackage{package1,package2,package3}
```

and you can have more than one `\usepackage` command. Some packages allow optional settings in square brackets. If you use these, you must give the package its own separate `\usepackage` command, like geometry shown below:

```
\documentclass[11pt,a4paper,oneside]{report}

\usepackage{pslatex,palatino,avant,graphicx,color}
\usepackage[margin=2cm]{geometry}

\begin{document}
\title{\color{red}Practical Typesetting}
\author{\color{blue}Name\\ Work}
\date{\color{green}December 2005}
\maketitle

\end{document}
```

Many packages can have additional formatting specifications in optional arguments in square brackets, in the same way as `geometry` does. Read the documentation for the package concerned to find out what can be done. You can pass several options together separated by a comma:

```
\usepackage[option1,option2,option3]{''package_name''}
```

5.3. The *document* environment

5.3.1. Top matter

At the beginning of most documents there will be information about the document itself, such as the title and date, and also information about the authors, such as name, address, email etc. All of this type of information within LaTeX is collectively referred to as *top matter* . Although never explicitly specified (there is no `\topmatter` command) you are likely to encounter the term within LaTeX documentation.

A simple example:

```
\documentclass[11pt,a4paper]{report}

\begin{document}
\title{How to Structure a LaTeX Document}
\author{Andrew Roberts}
\date{December 2004}
\maketitle
\end{document}
```

The `\title`, `\author`, and `\date` commands are self-explanatory. You put the title, author name, and date in curly braces after the relevant command. The title and author are usually compulsory (at least if you want LaTeX to write the title automatically); if you omit the `\date` command, LaTeX uses today's date by default. You always finish the top matter with the `\maketitle` command, which tells LaTeX that it's complete and it can typeset the title according to the information you have provided and the class (style) you are using. If you omit `\maketitle`, the titling will never be typeset (unless you write your own).

Here is a more complicated example:

```
\title{How to Structure a \LaTeX{} Document}
\author{Joe Bloggs\\
  School of Computing,\\
  University of Study,\\
  Books,\\
  United Readdom,\\
  RN 1234\\
  \texttt{jbloggs@latex.wizard}}
\date{\today}
\maketitle
```

as you can see, you can use commands as arguments of `\title` and the others. The double backslash (\\) is the LaTeX command for forced linebreak. LaTeX normally decides by itself where to break lines, and it's usually right, but sometimes you need to cut a line short, like here, and start a new one.

If there are two authors separate them with the `\and` command:

```
\title{Our Fun Document}
\author{Jane Doe \and John Doe}
\date{\today}
\maketitle
```

If you are provided with a class file from a publisher, or if you use the AMS article class (`amsart`), then you can use several different commands to enter author information. The email address is at the end, and the `\texttt` commands formats the email address using a mono-spaced font. The built-in command called `\today` will be replaced with the current date when processed by LaTeX. But you are free to put whatever you want as a date, in no set order. If braces are left empty, then the date is omitted.

Using this approach, you can create only basic output whose layout is very hard to change. If you want to create your title freely, see the Title Creation[2] section.

5.3.2. Abstract

As most research papers have an abstract, there are predefined commands for telling LaTeX which part of the content makes up the abstract. This should appear in its logical order, therefore, after the top matter, but before the main sections of the body. This command is available for the document classes *article* and *report* , but not *book* .

```
\documentclass{article}

\begin{document}

\begin{abstract}
Your abstract goes here...
...
\end{abstract}
...
\end{document}
```

By default, LaTeX will use the word ”Abstract” as a title for your abstract. If you want to change it into anything else, e.g. ”Executive Summary”, add the following line before you begin the abstract environment:

```
\renewcommand{\abstractname}{Executive Summary}
```

5.3.3. Sectioning commands

The commands for inserting sections are fairly intuitive. Of course, certain commands are appropriate to different document classes. For example, a book has chapters but an article doesn't. Here are some of the structure commands found in *simple.tex* .

```
\chapter{Introduction}
This chapter's content...

\section{Structure}
This section's content...

\subsection{Top Matter}
This subsection's content...

\subsubsection{Article Information}
This subsubsection's content...
```

2 Chapter 15 on page 187

Notice that you do not need to specify section numbers; LaTeX will sort that out for you. Also, for sections, you do not need to use `\begin` and `\end` commands to indicate which content belongs to a given block.

LaTeX provides 7 levels of depth for defining sections (see table below). Each section in this table is a subsection of the one above it.

Command	**Level**	**Comment**
`\part{''part''}`	-1	not in letters
`\chapter{''chapter''}`	0	only books and reports
`\section{''section''}`	1	not in letters
`\subsection{''subsection''}`	2	not in letters
`\subsubsection{''subsubsection''}`	3	not in letters
`\paragraph{''paragraph''}`	4	not in letters
`\subparagraph{''subparagraph''}`	5	not in letters

All the titles of the sections are added automatically to the table of contents (if you decide to insert one). But if you make manual styling changes to your heading, for example a very long title, or some special line-breaks or unusual font-play, this would appear in the Table of Contents as well, which you almost certainly don't want. LaTeX allows you to give an optional extra version of the heading text which only gets used in the Table of Contents and any running heads, if they are in effect. This optional alternative heading goes in [square brackets] before the curly braces:

```
\section[Effect on staff turnover]{An analysis of the
effect of the revised recruitment policies on staff
turnover at divisional headquarters}
```

Section numbering

Numbering of the sections is performed automatically by LaTeX, so don't bother adding them explicitly, just insert the heading you want between the curly braces. Parts get roman numerals (Part I, Part II, etc.); chapters and sections get decimal numbering like this document, and appendices (which are just a special case of chapters, and share the same structure) are lettered (A, B, C, etc.).

You can change the depth to which section numbering occurs, so you can turn it off selectively. By default it is set to 2. If you only want parts, chapters, and sections numbered, not subsections or subsubsections etc., you can change the value of the `secnumdepth` counter[3] using the `\setcounter` command, giving the depth level you wish. For example, if you want to change it to "1":

```
\setcounter{secnumdepth}{1}
```

A related counter is `tocdepth`, which specifies what depth to take the Table of Contents to. It can be reset in exactly the same way as `secnumdepth`. For example:

3 Chapter 24 on page 291

```
\setcounter{tocdepth}{3}
```

To get an unnumbered section heading which does not go into the Table of Contents, follow the command name with an asterisk before the opening curly brace:

```
\subsection*{Introduction}
```

All the divisional commands from `\part*` to `\subparagraph*` have this ”starred” version which can be used on special occasions for an unnumbered heading when the setting of `secnumdepth` would normally mean it would be numbered.

If you want the unnumbered section to be in the table of contents anyway, use the `\add-contentsline` command like this:

```
\section*{Introduction}
\addcontentsline{toc}{section}{Introduction}
```

Note that if you use PDF bookmarks you will need to add a phantom section so that bookmark will lead to the correct place in the document. The `\phantomsection` command is defined in the `hyperref` package, and is implemented normally as follows:

```
\phantomsection
\addcontentsline{toc}{section}{Introduction}
\section*{Introduction}
```

For chapters you will also need to clear the page (this will also correct page numbering in the ToC):

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{chapter}{Bibliography}
\bibliographystyle{unsrt}
\bibliography{my_bib_file}
```

The value where the section numbering starts from can be set with the following command:

```
\setcounter{section}{4}
```

The next section after this command will now be numbered 5.

For more details on counters, see the dedicated chapter[4].

Section number style

See Counters[5].

4 Chapter 24 on page 291
5 Chapter 24 on page 291

5.3.4. Ordinary paragraphs

Paragraphs of text come after section headings. Simply type the text and leave a blank line between paragraphs. The blank line means "start a new paragraph here": it does **not** mean you get a blank line in the typeset output. For formatting paragraph indents and spacing between paragraphs, refer to the Paragraph Formatting[6] section.

5.3.5. Table of contents

All auto-numbered headings get entered in the Table of Contents (ToC) automatically. You don't have to print a ToC, but if you want to, just add the command `\tableofcontents` at the point where you want it printed (usually after the Abstract or Summary).

Entries for the ToC are recorded each time you process your document, and reproduced the next time you process it, so you need to re-run LaTeX one extra time to ensure that all ToC pagenumber references are correctly calculated. We've already seen how to use the optional argument to the sectioning commands to add text to the ToC which is slightly different from the one printed in the body of the document. It is also possible to add extra lines to the ToC, to force extra or unnumbered section headings to be included.

The commands `\listoffigures` and `\listoftables` work in exactly the same way as `\tableofcontents` to automatically list all your tables and figures. If you use them, they normally go after the `\tableofcontents` command. The `\tableofcontents` command normally shows only numbered section headings, and only down to the level defined by the `tocdepth` counter, but you can add extra entries with the `\addcontentsline` command. For example if you use an unnumbered section heading command to start a preliminary piece of text like a Foreword or Preface, you can write:

```
\subsection*{Preface}
\addcontentsline{toc}{subsection}{Preface}
```

This will format an unnumbered ToC entry for "Preface" in the "subsection" style. You can use the same mechanism to add lines to the List of Figures or List of Tables by substituting `lof` or `lot` for `toc`. If the hyperref package is used and the link does not point to the correct chapter, the command `\phantomsection` in combination with `\clearpage` or `\cleardoublepage` can be used (see also Labels and Cross-referencing[7]):

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{chapter}{List of Figures}
\listoffigures
```

To change the title of the TOC, you have to paste this command `\renewcommand{\contentsname}{<New table of contents title>}` in your document preamble. The List of Figures (LoF) and List of Tables (LoT) names can be changed by replacing the `\contentsname` with `\listfigurename` for LoF and `\listtablename` for LoT.

6 Chapter 7.6.1 on page 85
7 Chapter 21.5 on page 274

Depth

The default ToC will list headings of level 3 and above. To change how deep the table of contents displays automatically the following command can be used in the preamble:

```
\setcounter{tocdepth}{4}
```

This will make the table of contents include everything down to paragraphs. The levels are defined above on this page. Note that this solution does not permit changing the depth dynamically.

You can change the depth of specific section type, which could be useful for PDF bookmarks (if you are using the `hyperref` package) :

```
\makeatletter
\renewcommand*{\toclevel@chapter}{-1} % Put chapter depth at the same level as
 \part.
\chapter{Epilogue}
\renewcommand*{\toclevel@chapter}{0} % Put chapter depth back to its default
 value.
\makeatother
```

In order to further tune the display or the numbering of the table of contents, for instance if the appendix should be less detailed, you can make use of the `tocvsec2` package (CTAN[8], doc[9]).

5.4. Book structure

The standard LaTeX `book` class follows the same layout described above with some additions. By default a book will be two-sided, *i.e.* left and right margins will change according to the page number parity. Furthermore current chapter and section will be printed in the header.

If you do not make use of chapters, it is barely useful to use the `book` class.

Additionally the class provides macros to change the formatting of some places of the document. We will give you some advice on how to use them properly.[10]

```
\begin{document}
\frontmatter

\maketitle

% Introductory chapters
\chapter{Preface}
% ...

\mainmatter
\chapter{First chapter}
% ...
```

8 `http://www.ctan.org/pkg/tocvsec2`
9 `http://mirror.ctan.org/macros/latex/contrib/tocvsec2/tocvsec2.pdf`
10 `http://tex.stackexchange.com/questions/20538/what-is-the-right-order-when-using-frontmatter-tableofcont`

```
\appendix
\chapter{First Appendix}

\backmatter
\chapter{Last note}
```

- The frontmatter chapters will not be numbered. Page numbers will be printed in roman numerals. Frontmatter is not supposed to have sections, since they will be number `0.n` because there is no chapter numbering. Check the Counters[11] chapter for a fix.
- The mainmatter chapters works as usual. The command resets the page numbering. Page numbers will be printed in arabic numerals.
- The `\appendix` macro can be used to indicate that following sections or chapters are to be numbered as appendices. Appendices can be used for the article class too:

```
\appendix
\section{First Appendix}
```

Only use the `\appendix` macro once for all appendices.

- The backmatter behaves like the frontmatter. It has the same issue with section numbering.

As a general rule you should avoid mixing the command order. Nonetheless all commands are optional, so you might consider using only a few.

Note that the special content like the table of contents is considered as an unnumbered chapter.

5.4.1. Page order

This is one traditional page order for books.

Frontmatter

1. Half-title
2. Empty
3. Title page
4. Information (copyright notice, ISBN, etc.)
5. Dedication if any, else empty
6. Table of contents
7. List of figures (can be in the backmatter too)
8. Preface chapter

Mainmatter

1. Main topic

Appendix

1. Some subordinate chapters

Backmatter

11 Chapter 24 on page 291

1. Bibliography
2. Glossary / Index

5.4.2. Introductory chapters with main page numbering

You may be tempted to put your introductory chapters in the main matter so that it follows the same numbering as the main chapters. This is not how the class was meant to be used, so you will run into an issue if you don’t want the chapter to be numbered.

```
\frontmatter
\maketitle
\tableofcontents

\mainmatter

%% WRONG!
\chapter*{Introduction}
Blah
\clearpage
Blah

\chapter{First one}
Blah
```

In the above code sample, the second page of the introduction will have *TABLE OF CONTENTS* printed in the header. This is because the starred `\chapter*` command does not set the leftmark -- see Page Layout[12]. And it will not be printed in the table of contents either.

The trick is to set the leftmark and the TOC manually:

```
\chapter*{Introduction}
\markboth{\MakeUppercase{Introduction}}{}
\addcontentsline{toc}{chapter}{Introduction}
% ...
```

TABLE OF CONTENTS is traditionally printed both left and right, but here we print it like other chapters, only right on even pages.

To make it more convenient, you might use a macro[13]:

```
\newcommand\intro[1]{
  \chapter*{#1}
  \markboth{\MakeUppercase{#1}}{}
  \addcontentsline{toc}{chapter}{#1}
}

%% ...

\intro{Introduction}
```

12 Chapter 16 on page 193
13 Chapter 51 on page 569

5.5. Special pages

Comprehensive papers often feature special pages at the end, like indices, glossaries and bibliographies. Since this is a quite complex topic, we will give you details in the dedicated part *Special Pages* .

5.5.1. Bibliography

Any good research paper will have a complete list of references. LaTeX has two ways of inserting your references into a document:

- you can embed them within the document itself. It's simpler, but it can be time-consuming if you are writing several papers about similar subjects so that you often have to cite the same books.
- you can store them in an external BibTeX file [14] and then link them via a command to your current document and use a Bibtex style[15] to define how they appear. This way you can create a small database of the references you might use and simply link them, letting LaTeX work for you.

To learn how to add a bibliography to your document, see the Bibliography Management[16] section.

5.6. Notes and references

ru:LaTeX/Структура документа[17]

14 `http://www.bibtex.org`

15 `http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html`

16 Chapter 38 on page 443

17 `http://ru.wikibooks.org/wiki/LaTeX%2F%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0%20%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0`

6. Text Formatting

This section will guide you through the formatting techniques of the text. *Formatting* tends to refer to most things to do with appearance, so it makes the list of possible topics quite eclectic: text style, spacing, etc. If *formatting* may also refer to paragraphs and to the page layout, we will focus on the customization of words and sentences for now.

A lot of formatting techniques are required to differentiate certain elements from the rest of the text. It is often necessary to add emphasis to key words or phrases. *Footnotes* are useful for providing extra information or clarification without interrupting the main flow of text. So, for these reasons, formatting is very important. However, it is also very easy to abuse, and a document that has been over-done can look and read worse than one with none at all.

LaTeX is so flexible that we will actually only skim the surface, as you can have much more control over the presentation of your document if you wish. Having said that, one of the purposes of LaTeX is to take away the stress of having to deal with the physical presentation yourself, so you need not get too carried away!

6.1. Spacing

6.1.1. 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, so the default line spread factor is 1.

The `setspace` package allows more fine-grained control over line spacing. To set ”one and a half” line spacing document-wide, but not where it is usually unnecessary (e.g. footnotes, captions):

```
\usepackage{setspace}
%\singlespacing
\onehalfspacing
%\doublespacing
%\setstretch{1.1}
```

To change line spacing within the document, the `setspace` package provides the environments `singlespace` , `onehalfspace` , `doublespace` and `spacing` :

```
This paragraph has \\ default \\ line spacing.

\begin{doublespace}
  This paragraph has \\ double \\ line spacing.
\end{doublespace}

\begin{spacing}{2.5}
  This paragraph has \\ huge gaps \\ between lines.
\end{spacing}
```

⚠ **Warning**

The line spacing value is contained in the `\baselineskip` length[1], but it is not recommended to change its value since it will have an impact on other types of content than paragraphs, which will result in an undesired effect.

6.1.2. Non-breaking spaces

This *essential* feature is a bit unknown to newcomers, although it is available on most WYSIWYG document processors. A non-breaking space between two tokens (e.g. words, punctuation marks) prevents the processors from inserting a line break between them. Besides a non-breaking space cannot be enlarged. It is very important for a consistent reading.

LaTeX uses the '~' symbol as a non-breaking space. You would usually use non-breaking spaces for punctuation marks in some languages, for units and currencies, for initials, etc. In French typography, you would put a non-breaking space before all two-parts punctuation marks.

Examples:

```
D.~\textsc{Knuth}
50~€
```

6.1.3. Space between words and sentences

To get a straight right margin in the output, LaTeX inserts varying amounts of space between the words. By default, it also inserts slightly more space at the end of a sentence. However, the extra space added at the end of sentences is generally considered typographically old-fashioned in English language printing. (The practice is found in nineteenth century design and in twentieth century typewriter styles.) Most modern typesetters treat the end of sentence space the same as the interword space. (See for example, Bringhurst's *Elements of Typographic Style* .) 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. Frenchspacing can be turned off later in your document via the `\nonfrenchspacing` command.

If an author wishes to use the wider end-of-sentence spacing, care must be exercised so that punctuation marks are not misinterpreted as ends of sentences. TeX assumes that sentences end with periods, question marks or exclamation marks. Although 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 that will not be enlarged. A tilde '~' character generates a non-breaking space. The command `\@` in front of a period specifies that this period terminates a sentence even when it follows an uppercase letter. (If you are using `\frenchspacing`, then none of these exceptions need be specified.)

6.1.4. Stretched spaces

You can insert a horizontal stretched space with `\hfill` in a line so that the rest gets ”pushed” toward the right margin. For instance this may be useful in the header.

```
Author Name \hfill \today
```

Similarly you can insert vertical stretched space with `\vfill`. It may be useful for special pages.

```
\maketitle
\vfill
\tableofcontents
\clearpage

\section{My first section}
% ...
```

See Lengths[2] for more details.

6.1.5. Manual spacing

The spaces between words and sentences, between paragraphs, sections, subsections, etc. is determined automatically by LaTeX. It is against LaTeX philosophy to insert spaces manually and will usually lead to bad formatting. Manual spacing is a matter of macro writing and package creation.

See Lengths[3] for more details.

6.2. Hyphenation

LaTeX hyphenates words whenever necessary. Hyphenation rules will vary for different languages. LaTeX only supports English by default, so if you want to have correct hyphenation rules for your desired language, see Internationalization[4].

2 Chapter 23 on page 283
3 Chapter 23 on page 283
4 Chapter 12 on page 133

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 characters that are considered to be normal letters by LaTeX. It is known that the hyphenation algorithm does not find all correct American English hyphenation points for several words. A log of known exceptions is published periodically in the *TUGboat* journal. (2012 list: `https://www.tug.org/TUGboat/tb33-1/tb103hyf.pdf)`

The hyphenation hints are stored for the language that 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}
```

The command `\-` inserts a discretionary hyphen into a word. This also becomes the only point where 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.

```
\begin{minipage}{2in}
I think this is: su\-per\-cal\-%
i\-frag\-i\-lis\-tic\-ex\-pi\-%
al\-i\-do\-cious
\end{minipage}
```

I think this is: supercalifragi-
listicexpialidocious

LaTeX does not hyphenate compound words that contain a dash[5]. There are two packages that can add back flexibility. The `hyphenat` package supplies the `\hyp` command. This command typesets the dash and then subjects the constituent words to automatic hyphenation. After loading the package:

```
\usepackage{hyphenat}
```

one should write, instead of electromagnetic-endioscopy:

```
electromagnetic\hyp{}endioscopy
```

The `extdash` package also offers features for controlling the hyphenation of compound words containing dashes — as opposed to the words themselves which it leaves to LaTeX. The `shortcuts` option enables a more compressed syntax:

5 `hyphenat` package documentation, p3

```
\usepackage[shortcuts]{extdash}
```

Typical usage is as follows, assuming the compressed syntax. In both cases, LaTeX can break and hyphenate the constituent words, but in the latter case, it will not break after the L:

```
electromagnetic\-/endioscopy
L\=/approximation
```

One or more words can be kept together on the **one line** with the standard LaTeX command:

```
\mbox{text}
```

This prevents hyphenation and causes its argument to be kept together under all circumstances. For example:

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

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

To avoid hyphenation altogether, the penalty for hyphenation can be set to an extreme value:

```
\hyphenpenalty=100000
```

You can change the degree to which LaTeX will hyphenate by changing the value of `\tolerance=1000` and `\hyphenpenalty=1000`. You'll have to experiment with the values to achieve the desired effect. A document which has a low tolerance value will cause LaTeX not to tolerate uneven spacing between words, hyphenating words more frequently than in documents with higher tolerances. Also note that using a higher text width will decrease the probability of encountering badly hyphenated word. For example adding

```
\usepackage{geometry}
```

will widen the text width and reduce the amount of margin overruns.

6.3. Quote-marks

LaTeX treats left and right quotes as different entities. For single quotes, ` (on American keyboards, this symbol is found on the tilde key (adjacent to the number 1 key on most keyboards) gives a left quote mark, and ' is the right. For double quotes, simply double the symbols, and LaTeX will interpret them accordingly. (Don't use the " for right double quotes: when the `babel` package is used for some languages (e.g. German), the " is redefined to produce an umlaut accent; using " for right double quotes will either lead to bad spacing or it being used to produce an umlaut). On British keyboards, ' ` ' is left of the ' 1 ' key and shares the key with ' ¬ ', and sometimes ' ¦ ' or ' | '. The apostrophe (') key is to the right of the colon/semicolon key and shares it with the ' @ ' symbol.

```
To `quote' in LaTeX
```

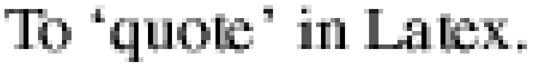


Figure 8

```
To ``quote'' in LaTeX
```

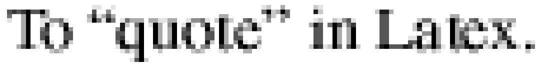

To “quote” in Latex.

Figure 9

```
To ``quote" in LaTeX
```

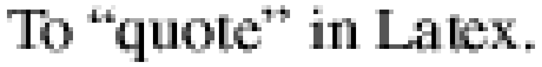

To “quote” in Latex.

Figure 10

```
To ,,quote'' in LaTeX
```

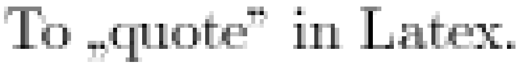

To „quote” in Latex.

Figure 11

```
,,German quotation marks``
```

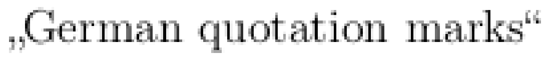

„German quotation marks“

Figure 12

```
<<French quotation marks>>
```

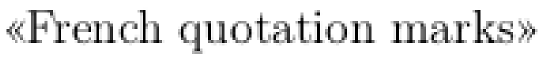
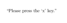

«French quotation marks»

Figure 13

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


“Please press the ‘x’ key.”

```
,,Proszę, naciśnij klawisz <<x>>''.
```

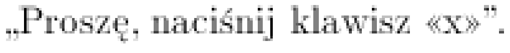

„Proszę, naciśnij klawisz «x»”.

Figure 15

The right quote is also used for apostrophe in LaTeX without trouble.

For left bottom quote and European quoting style you need to use T1 font encoding enabled by:

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

See Fonts[6] for more details on font encoding.

The package `csquotes` offers a multilingual solution to quotations, with integration to citation mechanisms offered by BibTeX. This package allows one for example to switch languages and quotation styles according to babel language selections.

6.4. Diacritics and accents

Most accents and diacritics may be inserted with direct keyboard input by configuring the preamble properly. For symbols unavailable on your keyboard, diacritics may be added to letters by placing special escaped metacharacters before the letter that requires the diacritic.

See Special Characters[7].

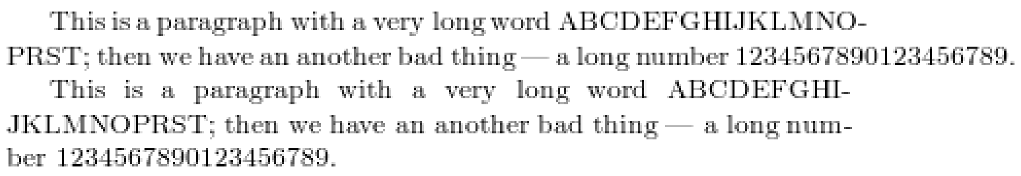
6.5. Margin misalignment and interword spacing

Some very long words, numbers or URLs may not be hyphenated properly and move far beyond the side margin. One solution for this problem is to use `sloppypar` environment,

6 Chapter 9 on page 95
7 Chapter 11 on page 123

which tells LaTeX to adjust word spacing less strictly. As a result, some spaces between words may be a bit too large, but long words will be placed properly.

```
This is a paragraph with
a very long word ABCDEFGHIJKLMNOPRST;
then we have another bad thing
--- a long number 1234567890123456789.

\begin{sloppypar}
This is a paragraph with
a very long word ABCDEFGHIJKLMNOPRST;
then we have an another bad thing
--- a long number 1234567890123456789.
\end{sloppypar}
```



This is a paragraph with a very long word ABCDEFGHIJKLMNO-
PRST; then we have an another bad thing — a long number 1234567890123456789.
This is a paragraph with a very long word ABCDEFGHI-
JKLMNOPRST; then we have an another bad thing — a long num-
ber 1234567890123456789.

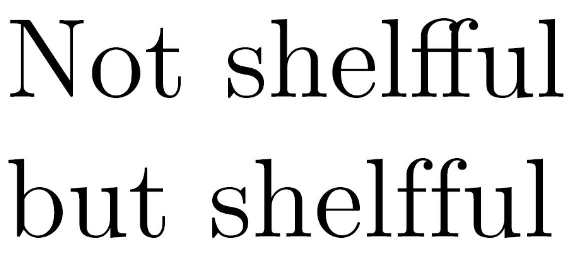
Figure 16 border

Another solution is to edit the text to avoid long words, numbers or URLs approaching the side margin.

6.6. Ligatures

Some letter combinations are typeset not just by setting the different letters one after the other, but by actually using special symbols (like ”ff”), called ligatures[8]. Ligatures can be prohibited by inserting `{}` or, if this does not work, `{\kern0pt}` between the two letters in question. This might be necessary with words built from two words. Here is an example:

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



Not shelfful
but shelfful

Figure 17

8 http://en.wikipedia.org/wiki/Typographical%20ligature

Ligatures can interfere with some text-search tools (a search for `"fi nally"` wouldn't find the string `"fi nally"`). The `\DisableLigatures` from the microtype package[9] can disable ligatures in the whole document to increase accessibility.

```
\usepackage{microtype}
\DisableLigatures{encoding = *, family = *}
```

Note that this will also disable ligatures such as "--" to "–", "---" to "—", etc.

If you are using XeLaTeX and OpenType fonts, the fontspec package allows for standard ligatures to be turned off as well as fancy swash ligatures to be turned on.

Another solution is to use the `cmap` package, which will help the reader to interpret the ligatures:

```
\usepackage[resetfonts]{cmap}
```

6.7. Slash marks

The normal typesetting of the `/` character in LaTeX does not allow following characters to be "broken" onto new lines, which often create "overfull" errors in output (where letters push off the margin). Words that use slash marks, such as "input/output" should be typeset as "`input\slash output`", which allow the line to "break" after the slash mark (if needed). The use of the `/` character in LaTeX should be restricted to units, such as "`mm/year`", which should not be broken over multiple lines.

A word after `/` or `\slash` is not automatically hyphenated. This is a similar problem to non-hyphenation of words with a dash described under Hyphenation[10]. One way to have both a line break and automatic hyphenation in both words is

```
input\slash\hspace{0pt}output
```

Both `/` and `\slash` can be used with a zero `\hspace` like this. `\slash` includes a penalty to make a line break there less desirable. This combination can be made into a new slash macro if desired. The `hyphenat` package includes an `\fshyp` which will add a hyphen after the slash like "input/- output" if the line breaks there.

6.8. Fonts

To change the font family, emphasize text, and other font-related issues, see Fonts[11].

9 `http://www.ctan.org/tex-archive/macros/latex/contrib/microtype/`
10 Chapter 6.2 on page 67
11 Chapter 9 on page 95

6.9. Formatting macros

Even if you can easily change the output of your fonts using those commands, you're better off not using 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]{\textit{#1}}

Do not \oops{enter} this room,
it's occupied by \oops{machines}
of unknown origin and purpose.
```

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

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

See Macros[12] for more details.

6.10. Text mode superscript and subscript

To superscript text in text-mode, you can use the `\textsuperscript{}` command. This allows you to, for instance, typeset 6th as 6^{th}:

```
Michelangelo was born on March 6\textsuperscript{th}, 1475.
```

Subscripting in text-mode is not supported by LaTeX alone; however, several packages allow the use of the `\textsubscript{}` command. For instance, bpchem[13], KOMA-Script2[14], and fixltx2e[15] all support this command. Of these, fixltx2e[16] is perhaps the most universal option since it is distributed with LaTeX and requires no additional packages to be implemented. Note that as of April 2015, fixltx2e[17] has been declared obsolete, and the `\textsubscript{}` command can no longer be used. changes[18] can be used to restore the `\textsubscript{}` command.

12 Chapter 51 on page 569
13 http://www.ctan.org/tex-archive/macros/latex/contrib/bpchem/
14 http://www.ctan.org/tex-archive/macros/latex/contrib/koma-script/
15 http://ctan.org/pkg/fixltx2e
16 http://ctan.org/pkg/fixltx2e
17 http://ctan.org/pkg/fixltx2e
18 http://changes.sourceforge.net/

```
% In your preamble, add:
\usepackage{fixltx2e}

% or
\usepackage{changes}

...

% In your document:
It is found that height\textsubscript{apple tree} is
different than height\textsubscript{orange tree}.
```

It is found that $\text{height}_{\text{apple tree}}$ is different than $\text{height}_{\text{orange tree}}$.

If you do not load a package that supports `\textsubscript{}`, the math mode must be used. This is easily accomplished in running text by bracketing your text with the `$` symbol. In math mode subscripting is done using the underscore: `_{}`.

For example, the formula for water is written as:

```
H$_2$O is the formula for water
```

H_2O is the formula for water

Note that in math mode text will appear in a font suitable for mathematical variables. In math mode, to generate Roman text, for example, one would use the `\mathrm` command:

```
This is $\mathrm{normal\ Roman\ and}_\mathrm{subscript\ Roman}$ text
```

This is $\mathrm{normal\ Roman\ and}_\mathrm{subscript\ Roman}$ text

Note the use of \<space> to insert a space in math mode.

Similarly, you can superscript using:

```
This is $\mathrm{normal\ Roman\ and}^\mathrm{superscript\ Roman}$ text
```

This is $\mathrm{normal\ Roman\ and}^\mathrm{superscript\ Roman}$ text

A very common use of subscripts within the text environment is to typeset chemical formulas. For these purposes, a highly recommended package is mhchem[19]. This package is easy to use and works with your text fonts (rather than math fonts). To insert a chemical formula, use `\ce{}` with the text-equivalent formula, for example:

```
% In your preamble, add:
\usepackage[version=3]{mhchem}
...

% In your document:
Ammonium sulphate is \ce{(NH4)2SO4}.
```

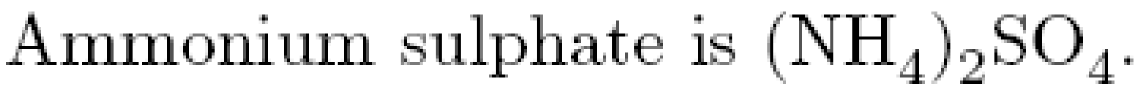

Ammonium sulphate is $(NH_4)_2SO_4$.

Figure 18

See also Chemical Graphics[20] for chemical symbols and formulas.

6.11. Text figures (”old style” numerals)

Many typographers prefer to use titling figures, sometimes called lining figures, when numerals are interspersed with full caps, when they appear in tables, and when they appear in equations, using text figures[21] elsewhere. LaTeX allows this usage through the `\oldstylenums{}` command:

```
\oldstylenums{1234567890}
```

Some fonts do not have text figures built in; the `textcomp` package attempts to remedy this by effectively generating text figures from the currently-selected font. Put `\usepackage{textcomp}` in your preamble. `textcomp` also allows you to use decimal points, properly formatted dollar signs, etc. within `\oldstylenums{}`.

One common use for text figures is in section, paragraph, and page numbers. These can be set to use text figures by placing some code in your preamble:

```
\usepackage{textcomp}

% Enclose everything in an \AtBeginDocument{}
\AtBeginDocument{%
  % Make \section{} use text figures
  \let\myTheSection\thesection
  \renewcommand{\thesection}{ \oldstylenums{\myTheSection} }

  % Make \paragraph{} use text figures
```

19 `http://www.ctan.org/tex-archive/macros/latex/contrib/mhchem/`
20 Chapter 30 on page 371
21 `http://en.wikipedia.org/wiki/Text%20figures`

```
  \let\myTheParagraph\theparagraph
  \renewcommand{\theparagraph}{ \oldstylenums{\myTheParagraph} }

  % Make the page numbers in text figures
  \let\myThePage\thepage
  \renewcommand{\thepage}{ \oldstylenums{\myThePage} }
}
```

Should you use additional sectioning or paragraphing commands, you may adapt the previous code listing to include them as well.

Note

A subsequent use of the `\pagenumbering` command, e.g., `\pagenumbering{arabic}`, will reset the `\thepage` command back to the original. Thus, if you use the `\pagenumbering` command in your document, be sure to reinstate your `\myThePage definition` from the code above:

```
...
\tableofcontents
\pagenumbering{roman}
\chapter{Preface}
...
\chapter{Introduction}
...
\pagenumbering{arabic}
% without this, the \thepage command will not be in oldstyle (e.g., in your
 Table of Contents}
\renewcommand{\thepage}{ \oldstylenums{\myThePage} }
\Chapter{Foo}
...
```

6.12. Dashes and hyphens

LaTeX knows four kinds of dashes: a hyphen[22] (-), en dash[23] (–), em dash[24] (—), or a minus sign[25] (−). 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:

```
 Hyphen: daughter-in-law, X-rated\\
 En dash: pages 13--67\\
 Em dash: yes---or no? \\
 Minus sign: $0$, $1$ and $-1$
```

22 http://en.wikipedia.org/wiki/hyphen
23 http://en.wikipedia.org/wiki/Dash%23En%20dash
24 http://en.wikipedia.org/wiki/Dash%23Em%20dash
25 http://en.wikipedia.org/wiki/Plus%20and%20minus%20signs%23Minus%20sign

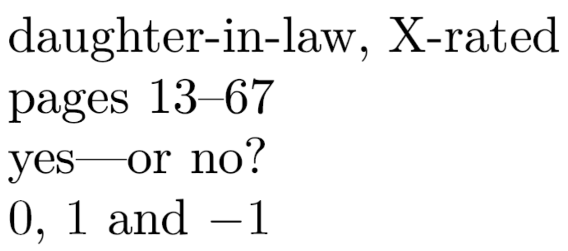
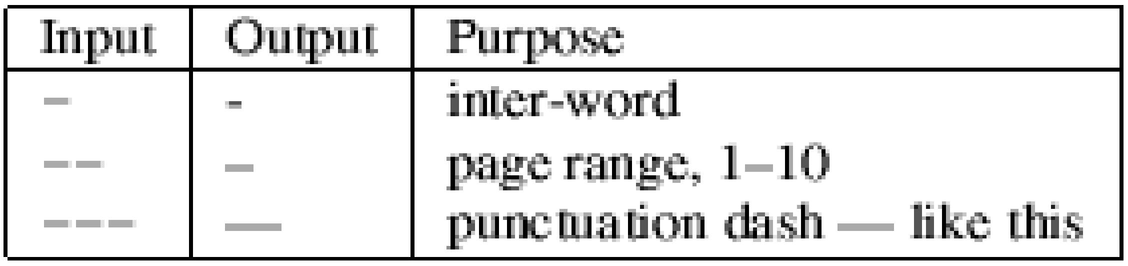

daughter-in-law, X-rated
pages 13–67
yes—or no?
0, 1 and −1

Figure 19

The names for these dashes are: '-'(-) hyphen , '--'(–) en-dash , '---'(—) em-dash and '−'(−) minus sign. They have different purposes:



Input	Output	Purpose
-	-	inter-word
--	–	page range, 1–10
---	—	punctuation dash — like this

Figure 20

Use `\hyp{}` macro from `hyphenat` package instead of hyphen if you want LaTeX to break compound words between lines.

The commands `\textendash` and `\textemdash` are also used to produce en-dash (–), and em-dash (—), respectively.

6.13. Ellipsis (...)

A sequence of three dots is known as an *ellipsis*[26] , which is commonly used to indicate omitted text. 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. Instead, there is a special command for these dots. It is called `\ldots`:

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

26 http://en.wikipedia.org/wiki/Ellipsis

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

Figure 21

Alternatively, you can use the `\textellipsis` command which allows the spacing between the dots to vary.

6.14. Ready-made strings

There are some very simple LaTeX commands for typesetting special text strings:

Command	Example	Description
`\today`	May 31, 2006	Current date
`\TeX`	TEX	Your favorite typesetter
`\LaTeX`	LATEX	The Name of the Game
`\LaTeXe`	LATEX 2$_\varepsilon$	The current incarnation

Figure 22

6.15. Notes and References

This page uses material from Andy Roberts' Getting to grips with LaTeX with permission from the author.

ru:LaTeX/Форматирование текста[27]

27 `http://ru.wikibooks.org/wiki/LaTeX%2F%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%B0`

7. Paragraph Formatting

Altering the paragraph formatting is rarely necessary in academic writing. It is primarily used for formatting text in floats or for more exotic documents.

7.1. Paragraph alignment

Paragraphs in LaTeX are usually fully justified, *i.e.* flush with both the left and right margins. For whatever reason, should you wish to alter the justification of a paragraph, there are three environments at hand, and also LaTeX command equivalents.

Alignment	**Environment**	**Command**
Left justified	`flushleft`	`\raggedright`
Right justified	`flushright`	`\raggedleft`
Center	`center`	`\centering`

All text between the `\begin` and `\end` of the specified environment will be justified appropriately. The commands listed are for use within other environments. For example, `p` (paragraph) columns in `tabular`.

⚠ **Warning**

There is no way (in standard LaTeX) to set full justification explicitly. It means that if you do not enclose the previous 3 commands into a group, the rest of the document will be affected. So the right way of doing this with commands is

```
{\raggedleft{}Some text flushed right.}
```

However, if you *really* need to disable one of the above commands locally (for example because you have to use some broken package), you can use the command `\justifying` from package `ragged2e`.

7.2. Paragraph indent and break

By default, the first paragraph after a heading follows the standard Anglo-American publishers' practice of no indentation. The size of subsequent paragraph indents is determined by a parameter called `\parindent`. The default length that this constant holds is set by the document class that you use. It is possible to override it by using the `\setlength` command. This will set paragraph indents to 1cm:

```
\setlength{\parindent}{1cm} % Default is 15pt.
```

Whitespace in LaTeX can also be made flexible (what Lamport calls "rubber" lengths). This means that values such as extra vertical space inserted before a paragraph `\parskip` can have a default dimension plus an amount of expansion minus an amount of contraction. This is useful on pages in complex documents where not every page may be an exact number of fixed-height lines long, so some give-and-take in vertical space is useful. You specify this in a `\setlength` command like this:

```
\setlength{\parskip}{1cm plus4mm minus3mm}
```

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

```
\indent
```

at the beginning of the paragraph. Obviously, this will only have an effect when `\parindent` is not set to zero. If you want to indent the beginning of every section, you can use the `indentfirst` package: once loaded, the beginning of any chapter/section is indented by the usual paragraph indentation.

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.

Be careful, however, if you decide to set the indent to zero, then it means you will need a vertical space between paragraphs in order to make them clear. The space between paragraphs is held in `\parskip`, which could be altered in a similar fashion as above. However, this parameter is used elsewhere too, such as in lists, which means you run the risk of making various parts of your document look very untidy by changing this setting. If you want to use the style of having no indentation with a space between paragraphs, use the `parskip` package, which does this for you, while making adjustments to the spacing of lists and other structures which use paragraph spacing, so they don't get too far apart. If you want both indent and break, use

```
\usepackage{parskip}
\setlength{\parindent}{15pt}
```

To indent subsequent lines of a paragraph, use the TeX command `\hangindent`. (While the default behaviour is to apply the hanging indent after the first line, this may be changed with the `\hangafter` command.) An example follows.

```
\hangindent=0.7cm This paragraph has an extra indentation at the left.
```

The TeX commands `\leftskip` and `\rightskip` add additional space to the left and right sides of each line, allowing the formatting for subsequent paragraphs to differ from the overall document margins. This space is in addition to the indentation added by `\parindent` and `\hangindent`.

To change the indentation of the last line in a paragraph, use the TeX command `\parfill-skip`.

7.3. \paragraph line break

Default style for `\paragraph` may seem odd in the first place, as it writes the following text next to the title. If you do not like it, use a class other than the traditional article/book, or use ConTeXt or PlainTeX. Hacking of the class in use is really not the way LaTeX is intended to be used, and you may encounter a lot of frustrating issues.

Anyway, let's analyse the problem. If you add a manual line break with \\, LaTeX will complain that

```
There's no line here to end.
```

Simply adding an empty space will do it:

```
\paragraph{Title} \hspace{0pt} \\
Text...
```

Alternatively you can use the shorter, yet not completely equivalent syntax:

```
\paragraph{Title} ~\\
Text...
```

7.4. Line spacing

To change line spacing in the whole document use the command `\linespread` covered in Text Formatting[1].

Alternatively, you can use the `\usepackage{setspace}` package, which is also covered in Text Formatting[2]. This package provides the commands `\doublespacing`, `\onehalfspacing`, `\singlespacing` and `\setstretch{baselinestretch}`, which will specify the line spacing for all sections and paragraphs until another command is used. Furthermore, the package provides the following environments in order to change line spacing within the document but not document-wide:

- `doublespace`: lines are double spaced;
- `onehalfspace`: line spacing set to one-and-half spacing;
- `singlespace`: normal line spacing;
- `spacing`: customizable line spacing, e.g. `\begin{spacing}{\baselinestretch} ... \end{spacing}`.

See the section on customizing lists[3] for information on how to change the line spacing in lists.

1 Chapter 6.1.1 on page 65
2 Chapter 6.1.1 on page 65
3 Chapter 10.3 on page 114

7.5. Manual breaks

LaTeX takes care of formatting, breaks included. You should avoid manual breaking as much as possible, for it could lead to very bad formatting.

Controlling the breaks should be reserved to macro and package writers. Here follows a quick reference.

`\newline`	Breaks the line at the point of the command.
`\\`	Breaks the line at the point of the command; it is usually a shorter version of the previous command, but LaTeX sometimes redefines it for several environments. This command also features the vertical space as optional parameter.
`\\*`	Breaks the line at the point of the command and additionally prohibits a page break after the forced line break. This command also features the vertical space as optional parameter.
`\linebreak[number]`	Breaks the line at the point of the command. The *number* you provide as an argument represents the priority of the command in a range from 0 (it will be easily ignored) to 4 (do it anyway). LaTeX will try to produce the best line breaks possible. If it cannot, it will decide whether including the linebreak or not according to the priority you have provided.
`\break` (TeX)	Breaks the line without filling the current line. This will result in an underful badness if you do not fill the line yourself, *i.e.* `...\hfill\break ...`. Actually `\hfill\break` produces the same as `\newline` and `\\`.
`\par` (TeX)	Starts a new paragraph. It is a horizontal mode command, so you can only use it in a paragraph.

The page breaks are covered in Page Layout[4]. More details on manual spaces between paragraphs (such as `\bigskip`) can be found in Lengths[5].

7.6. Special paragraphs

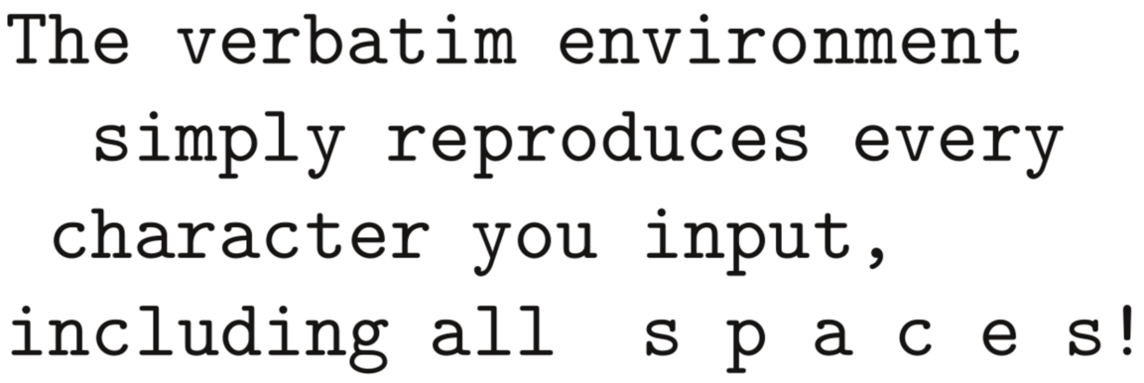
7.6.1. Verbatim text

There are several ways to introduce text that won't be interpreted by the compiler. If you use the `verbatim` environment, everything input between the *begin* and *end* commands are processed as if by a typewriter. All spaces and new lines are reproduced as given, and the text is displayed in an appropriate fixed-width font. Any LaTeX command will be ignored

4 Chapter 16.10 on page 208
5 Chapter 23 on page 283

and handled as plain text. This is ideal for typesetting program source code. Here is an example:

```
\begin{verbatim}
The verbatim environment
  simply reproduces every
 character you input,
including all  s p a c e s!
\end{verbatim}
```

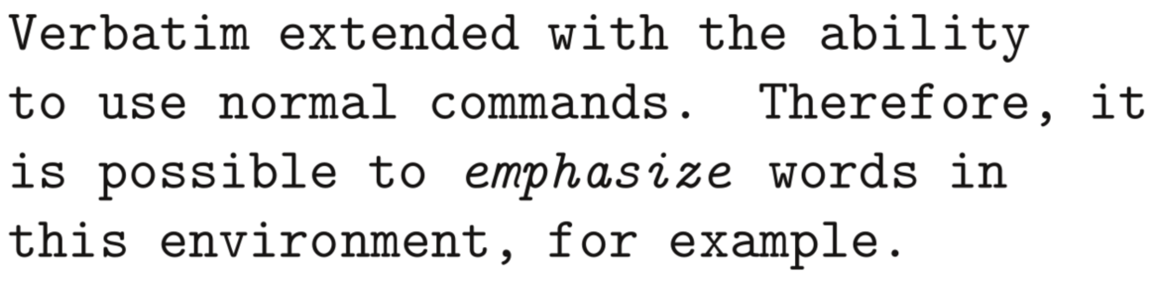


```
The verbatim environment
  simply reproduces every
 character you input,
including all  s p a c e s!
```

Figure 23

Note: once in the `verbatim` environment, the only command that will be recognized is `\end{verbatim}`. Any others will be output. The font size in the verbatim environment can be adjusted by placing a font size command[6] before `\begin{verbatim}`. If this is a problem, you can use the `alltt` package instead, providing an environment with the same name:

```
\begin{alltt}
Verbatim extended with the ability
to use normal commands.  Therefore, it
is possible to \emph{emphasize} words in
this environment, for example.
\end{alltt}
```



```
Verbatim extended with the ability
to use normal commands.  Therefore, it
is possible to emphasize words in
this environment, for example.
```

Figure 24

6 Chapter 7.6.1 on page 85

Remember to add `\usepackage{alltt}` to your preamble to use it though! Within the `alltt` environment, you can use the command `\normalfont` to get back the normal font. To write equations within the `alltt` enviroment, you can use `\(` and `\)` to enclose them, instead of the usual `$`.

When using `\textbf{}` inside the `alltt` enviroment, note that the standard font has no bold TT font. Txtfonts has bold fonts: just add `\renewcommand{\ttdefault}{txtt}` after `\usepackage{alltt}`.

If you just want to introduce a short verbatim phrase, you don't need to use the whole environment, but you have the `\verb` command:

```
\verb+my text+
```

The first character following `\verb` is the delimiter: here we have used "+", but you can use any character you like except *; `\verb` will print verbatim all the text after it until it finds the next delimiter. For example, the code:

```
\verb;\textbf{Hi mate!};
```

will print `\textbf{Hi mate!}`, ignoring the effect `\textbf` should have on text.

For more control over formatting, however, you can try the `fancyvrb` package, which provides a `Verbatim` environment (note the capital letter) which lets you draw a rule round the verbatim text, change the font size, and even have typographic effects inside the `Verbatim` environment. It can also be used in conjunction with the `fancybox` package and it can add reference line numbers (useful for chunks of data or programming), and it can even include entire external files.

To use verbatim in beamer, the frame needs to be make fragile: \begin{frame}[fragile] .

Typesetting URLs

One of either the `hyperref` or `url` packages provides the `\url` command, which properly typesets URLs, for example:

```
Go to \url{http://www.uni.edu/~myname/best-website-ever.html} for my website.
```

will show this URL exactly as typed (similar to the `\verb` command), but the `\url` command also performs a hyphenless break at punctuation characters (only in PDFLaTeX, not in plain LaTeX+ dvips). It was designed for Web URLs, so it understands their syntax and will never break midway through an unpunctuated word, only at slashes and full stops. Bear in mind, however, that spaces are forbidden in URLs, so using spaces in `\url` arguments will fail, as will using other non-URL-valid characters.

When using this command through the `hyperref` package, the URL is "clickable" in the PDF document, whereas it is not linked to the web when using only the `url` package. Also when using the `hyperref` package, to remove the border placed around a URL, insert `pdfborder = {0 0 0 0}` inside the `\hypersetup{}`. (Alternately `pdfborder = {0 0 0}` might work if the four zeroes do not.)

You can put the following code into your preamble to change the style, how URLs are displayed to the normal font:

```
\urlstyle{same}
```

See also Hyperlinks[7]

Listing environment

This is also an extension of the verbatim environment provided by the `moreverb` package. The extra functionality it provides is that it can add line numbers along side the text. The command: `\begin{listing}[step]{first line}`. The mandatory *first line* argument is for specifying which line the numbering shall commence. The optional *step* is the step between numbered lines (the default is 1, which means every line will be numbered).

To use this environment, remember to add `\usepackage{moreverb}` to the document preamble.

7.6.2. Multiline comments

As we have seen, the only way LaTeX allows you to add comments is by using the special character `%`, that will comment out all the rest of the line after itself. This approach is really time-consuming if you want to insert long comments or just comment out a part of your document that you want to improve later, unless you're using an editor[8] that automates this process. Alternatively, you can use the `verbatim` package, to be loaded in the preamble as usual:

```
\usepackage{verbatim}
```

(you can also use the `comment` package instead) you can use an environment called `comment` that will comment out everything within itself. Here is an example:

```
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. You may be wondering, why should I load a package called `verbatim` to have the possibility to add comments? The answer is straightforward: commented text is interpreted by the compiler

7 Chapter 20.3 on page 258
8 Chapter 1.5 on page 9

just like verbatim text, the only difference is that verbatim text is introduced within the document, while the comment is just dropped.

Alternatively, you can define a `\comment{}` command, by adding the following to the document's preamble:

```
\newcommand{\comment}[1]{}
```

Then, to comment out text, simply do something like this:

```
\comment{This is a long comment and can extend over multiple lines, etc.} But it
 won't show.
```

But it won't show.

This approach can, however, produce unwanted spaces in the document, so it may work better to use

```
\newcommand{\comment}[2]{#2}
```

Then if you supply only one argument to `\comment{}`, this has the desired effect without producing extra spaces.

Another drawback is that content is still parsed and possibly expanded, so you cannot put anything you want in it (such as LaTeX commands).

7.6.3. Skipping parts of the source

A more robust way of making the TeX engine skip some part of the source is to use the TeX `\iffalse`-conditional. The typical use is

```
This we want to keep

\iffalse % ----- START THE CUT ---------

But this part
$$\int_{-\infty}^\infty\mathrm{d}x\,x^{-2}$$
we want to skip

\fi % ---------- END THE CUT -----------

Here it begins again
```

This we want to keep
Here it begins again

The `\iffalse`-conditional is always false.

7.6.4. Quoting text

LaTeX provides several environments for quoting text; they have small differences and they are aimed for different types of quotations. All of them are indented on either margin, and you will need to add your own quotation marks if you want them. The provided environments are:

`quote`

for a short quotation, or a series of small quotes, separated by blank lines.

`quotation`

for use with longer quotations, of more than one paragraph, because it indents the first line of each paragraph.

`verse`

is for quotations where line breaks are important, such as poetry. Once in, new stanzas are created with a blank line, and new lines within a stanza are indicated using the newline command,

```
\\
```

. If a line takes up more than one line on the page, then all subsequent lines are indented until explicitly separated with

```
\\
```

.

7.6.5. Abstracts

In scientific publications it is customary to start with an abstract which gives the reader a quick overview of what to expect. See Document Structure[9].

7.7. Notes and References

This page uses material from Andy Roberts' Getting to grips with LaTeX with permission from the author.

9 Chapter 5.3.2 on page 56

8. Colors

Adding colors to your text is supported by the `color` [1] package. Using this package, you can set the font color, text background, or page background. You can choose from 230 predefined colors or can define your own colors using RGB, Hex, or CMYK codes. Mathematical formulas can also be colored.

8.1. Adding the color package

To make use of these color features the color package must be inserted into the preamble.

```
\usepackage{color}
```

Alternatively, one can write:

```
\usepackage[usenames,dvipsnames,svgnames,table]{xcolor}
```

The `\usepackage` is obvious, but the initialization of additional commands like `usenames` allows you to use names of the default colors, the same 16 base colors as used in HTML. The `dvipsnames` allows you access to more colors, another 64, and `svgnames` allows access to about 150 colors. The initialization of ”table” allows colors to be added to tables by placing the color command just before the table. The package loaded here is the `xcolor` [2] package.

If you need more colors, then you may also want to look at adding the `x11names` to the initialization section as well, this offers more than 300 colors, but you need to make sure your xcolor package is the most recent you can download.

8.2. Entering colored text

The simplest way to type colored text is by:

```
\textcolor{declared-color}{text}
```

where `declared-color` is a color that was defined before by `\definecolor`.

Another possible way by

```
{\color{declared-color} some text}
```

1 http://www.ctan.org/pkg/color
2 http://www.ctan.org/pkg/xcolor

that will switch the standard text color to the color you want. It will work until the end of the current TeX group. For example:

```
\emph{some black text, \color{red} followed by a red fragment}, going black
 again.
```

some black text, followed by a red fragment, going black again.

Figure 25

The difference between `\textcolor` and `\color` is the same as that between `\texttt` and `\ttfamily`, you can use the one you prefer. The `\color` environment allows the text to run over multiple lines and other text environments whereas the text in `\textcolor` must all be one paragraph and not contain other environments.

You can change the background color of the whole page by:

```
\pagecolor{declared-color}
```

8.3. Entering colored background for the text

```
\colorbox{declared-color}{text}
```

If the background color and the text color is changed, then:

```
\colorbox{declared-color1}{\color{declared-color2}text}
```

There is also \fcolorbox to make framed background color in yet another color:

```
\fcolorbox{declared-color-frame}{declared-color-background}{text}
```

8.4. Predefined colors

The predefined color names are

```
white, black, red, green, blue, cyan, magenta, yellow.
```

There may be other pre-defined colors on your system, but these should be available on all systems.

If you would like a color not pre-defined, you can use one of the 68 dvips colors, or define your own. These options are discussed in the following sections

8.4.1. The 68 standard colors known to dvips

Invoke the package with the usenames and dvipsnames option. If you are using `tikz` or `pstricks` package you must declare the xcolor package before that, otherwise it will not work.

```
\usepackage[usenames,dvipsnames]{xcolor}
```

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

8.5. Defining new colors

If the predefined colors are not adequate, you may wish to define your own.

8.5.1. Place

Define the colors in the *preamble* of your document. (Reason: do so in the preamble, so that you can already refer to them in the preamble, which is useful, for instance, in an argument of another package that supports colors as arguments, such as the listings[3] package.)

8.5.2. Method

You need to include the `xcolor` package in your preamble to define new colors. In the abstract, the colors are defined following this scheme:

```
\definecolor{name}{model}{color-spec}
```

where:

- *name* is the name of the color; you can call it as you like
- *model* is the way you *describe* the color, and is one of *gray* , *rgb* , *RGB* , *HTML* , and *cmyk* .
- *color-spec* is the description of the color

8.5.3. Color Models

Among the models you can use to describe the color are the following (several more are described in the xcolor manual[4]):

Color Models			
Model	**Description**	**Color Specification**	**Example**
`gray`	Shades of gray (0-1)	Just one number between 0 (black) and 1 (white), so 0.95 will be very light gray, 0.30 will be dark gray.	`\definecolor{light-gray}{gray}{0.95}`
`rgb`	Red, Green, Blue (0-1)	Three numbers given in the form *red,green,blue* ; the quantity of each color is represented with a number between 0 and 1.	`\define-color{orange}{rgb}{1,0.5,0}`
`RGB`	Red, Green, Blue (0-255)	Three numbers given in the form *red,green,blue* ; the quantity of each color is represented with a number between 0 and 255.	`\define-color{orange}{RGB}{255,127,0}`
`HTML`	Red, Green, Blue (00-FF)	Six hexadecimal numbers given in the form *RRGGBB* ; similar to what is used in HTML.	`\define-color{orange}{HTML}{FF7F00}`

3 Chapter 32 on page 393
4 `http://mirror.ctan.org/macros/latex/contrib/xcolor/xcolor.pdf`

Color Models			
Model	**Description**	**Color Specification**	**Example**
`cmyk`	Cyan, Magenta, Yellow, Black (0-1)	Four numbers given in the form *cyan,magenta,yellow,black* ; the quantity of each color is represented with a number between 0 and 1.	`\define-color{orange}{cmyk}{0,0.5,1,0}`

8.5.4. Examples

To define a new color, follow the following example, which defines orange for you, by setting the red to the maximum, the green to one half (0.5), and the blue to the minimum:

```
\definecolor{orange}{rgb}{1,0.5,0}
```

The following code should give a similar results to the last code chunk.

```
\definecolor{orange}{RGB}{255,127,0}
```

If you loaded the `xcolor` package, you can define colors upon previously defined ones.

The first specifies 20 percent blue and 80 percent white; the second is a mixture of 20 percent blue and 80 percent black; and the last one is a mixture of (20*0.3) percent blue, ((100-20)*0.3) percent black and (100-30) percent green.

```
\color{blue!20}
\color{blue!20!black}
\color{blue!20!black!30!green}
```

`xcolor` also feature a handy command to define colors from color mixes:

```
\colorlet{notgreen}{blue!50!yellow}
```

8.5.5. Using color specifications directly

Normally one would predeclare all the colors as above, but sometimes it is convenient to directly use a color without naming it first. To achieve this, `\color` and `\textcolor` have an alternative syntax specifying the model in square brackets, and the color specification in curly braces. For example:

```
{\color[rgb]{1,0,0} This text will appear red-colored}
\textcolor[rgb]{0,1,0}{This text will appear green-colored}
```

8.5.6. Creating / Capturing colors

You may want to use colors that appear on another document, web pages, pictures, etc. Alternatively, you may want to play around with rgb values to create your own custom colors.

Image processing suites like the free GIMP[5] suite for Linux/Windows/Mac offer color picker facilities to capture any color on your screen or synthesize colors directly from their respective rgb / hsv / hexadecimal values.

Smaller, free utilities also exist:

- Linux/BSD: The gcolor2[6] tool (usually also available in repositories)
- Microsoft Windows: The open-source Color Selector[7] tool.
- Apple Macs: Hex Color Picker[8] for creating custom colors and the built-in DigitalColor Meter[9] for capturing colors on screen.
- Online utilities: See here for a Wikipedia article with several external links[10]

8.5.7. Spot colors

Spot colors are customary in printing. They usually refer to pre-mixed inks based on a swatchbook (like Pantone, TruMatch or Toyo). The package `colorspace` extends xcolor to provide real spot colors. They are defined with, say:

```
\definespotcolor{mygreen}{PANTONE 7716 C}{.83, 0, .40, .11}
```

8.6. Sources

- The xcolor manual[11]
- The color package documentation [12]

5 http://www.gimp.org/downloads/
6 http://gcolor2.sourceforge.net/
7 http://colorselector.sourceforge.net/
8 http://wafflesoftware.net/hexpicker/
9 http://www.apple.com/uk/osx/apps/all.html#colormeter
10 http://en.wikipedia.org/wiki/Color_tool
11 http://mirror.ctan.org/macros/latex/contrib/xcolor/xcolor.pdf
12 http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf

9. Fonts

Fonts are a complex topic. For common documents, only Font families[1], Emphasizing text[2], and Font encoding[3] are really needed. The other sections are more useful to macro writers or for very specific needs.

9.1. Introduction

The digital fonts have a long and intricate history. See Adobe Font Metrics[4] for some more details.

Originally TeX was conceived to use its own font system, MetaFont, designed by D. Knuth. The default font family for TeX and friends is called Computer Modern. These high quality fonts are scalable, and have a wide range of typographical fine tuning capabilities.

Standard `tex` compilers will let you use other fonts. There are many different font types, such as PostScript Type1/Type3 fonts and bitmap fonts. Type1 are outline fonts (vector graphics) which are commonly used by `pdftex` . Bitmap fonts are raster graphics, and usually have very poor quality, which can easily be seen when zooming or printing a document. Type3 is a superset of Type1 and has more functionalities from Postscript, such as embedding raster graphics. In the TeX world, Type3 fonts are often used to embed bitmap fonts.

It should be noticed that fonts get generated the first time they are required, hence the long compilation time.

However, MetaFont is internally a quite complex font system, and the most popular font systems as of this day are Truetype[5] font (ttf) and OpenType[6] font (otf). With modern TeX compilers such as `xetex` and `luatex` it is possible to make use of such fonts in LaTeX documents. If you want/have to stick with the standard compilers, the aforementioned font types must first be converted and made available to LaTeX (*e.g.* converted to Type1 fonts). The external links section below has some useful resources.

In LaTeX, there are many ways to specify and control fonts. It is a very complex matter in typography.

1 Chapter 9.2 on page 96
2 Chapter 9.4 on page 98
3 Chapter 9.5 on page 99
4 `http://en.wikipedia.org/wiki/Adobe%20Font%20Metrics`
5 `http://en.wikipedia.org/wiki/TrueType`
6 `http://en.wikipedia.org/wiki/OpenType`

9.2. Font families

There are many font families e.g. Computer Modern, Times, Arial and Courier. Those families can be grouped into three main categories: roman (rm) or serif, sans serif (sf) and monospace (tt) (see Typeface[7] for more details). Each font family comes with the default design which falls into one of those categories; however, it is interchangeable among them. Computer Modern Roman is the default font family for LaTeX. Fonts in each family also have different properties (size, shape, weight, etc.). Families are meant to be consistent, so it is highly discouraged to change fonts individually rather than the whole family.

The three families are defined by their respective variables:

- `\rmdefault`
- `\sfdefault`
- `\ttdefault`

The default family is contained in the `\familydefault` variable, and it is meant to have one of the three aforementioned variables as value. The default is defined like the following assignment:

```
\renewcommand*{\familydefault}{\sfdefault}
```

This will turn all the part of the document using the default font to the default sans serif, which is Computer Modern Sans Serif if you did not change the default font.

Changing font families usually works in two steps:

1. First specify which family you want to change (rm, sf or tt).
2. Second specify the new default family if it is not rm.

Mathematical fonts is a more complex matter. Fonts may come with a package that will take care of defining all three families plus the math fonts. You can do it by yourself, in which case you do not have to load any package.

Below is an example[8]that demonstrates how to change a specific family.

```
% Default font (\familydefault = \rmdefault = Computer Modern Roman)
Lorem ipsum dolor sit amet, consectitur adipiscing elit.

% Palatino font (ppl must be installed).
\renewcommand*\rmdefault{ppl}
Lorem ipsum dolor sit amet, consectitur adipiscing elit.

% Iwona font (iwona must be installed).
\renewcommand*\rmdefault{iwona}
Lorem ipsum dolor sit amet, consectitur adipiscing elit.
```

7 `http://en.wikipedia.org/wiki/Typeface`
8 found at the Google discussion group *latexlovers*

Lorem ipsum dolor sit amet, consectitur adipiscing elit.
Lorem ipsum dolor sit amet, consectitur adipiscing elit.
Lorem ipsum dolor sit amet, consectitur adipiscing elit.

Figure 26

The three default family font variables and the `\familydefault` variable should not be confused with their respective switch:

- `\normalfont`
- `\rmfamily`
- `\sffamily`
- `\ttfamily`

9.3. Available LaTeX Fonts

To choose a font of your liking, please visit `http://www.tug.dk/FontCatalogue/`. Here are some common examples.

Below are some fonts which are installed by default.

Serif Fonts

Abbreviation	**Font Name**
cmr	Computer Modern Roman (default)
lmr	Latin Modern Roman
pbk	Bookman
bch	Charter
pnc	New Century Schoolbook
ppl	Palatino
ptm	Times

Sans Serif Fonts

Abbreviation	**Font Name**
cmss	Computer Modern Sans Serif (default)
lmss	Latin Modern Sans Serif
pag	Avant Garde
phv	Helvetica

Typewriter Fonts

Abbreviation	**Font Name**
cmtt	Computer Modern Typewriter (default)
lmtt	Latin Modern

Abbreviation	Font Name
pcr	Courier

Furthermore, the Bera Mono[9] (BitStream Vera Mono) and LuxiMono[10] fonts were designed to look good when used in conjunction with the Computer Modern serif font.

```
\usepackage[scaled=0.85]{beramono}
```

Cursive Fonts

Since LaTeX has no generic family group for cursive fonts, these fonts are usually assigned to the roman family.

Abbreviation	Font Name
pzc	Zapf Chancery

Mathematical Formula Fonts

Abbreviation	Font Name
cmm	Computer Modern (math italic)
cmsy	Computer Modern (math symbols)
zplm	Palatino (math)

9.4. Emphasizing text

⚠ **Warning**

Do not overuse emphasis in your paragraphs. Emphasis should be reserved for only key terms or other particularly important concepts in a text, and bold text especially used minimally.

In order to add some emphasis to a word or a phrase, the simplest way is to use the `\emph{text}` command, which usually italicizes the text. Italics may be specified explicitly with `\textit{text}`.

```
I want to \emph{emphasize} a word.
```

9 http://www.ctan.org/tex-archive/fonts/bera/
10 http://www.ctan.org/tex-archive/fonts/LuxiMono/

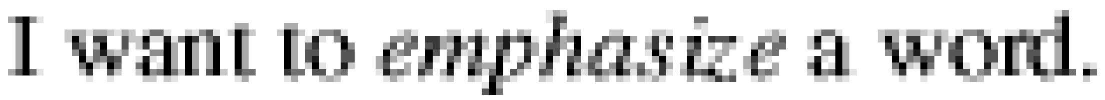


Figure 27

Note that the `\emph` command is dynamic: if you emphasize a word which is already in an emphasized sentence, it will be reverted to the upright font.

```
\emph{In this emphasized sentence, there is an emphasized \emph{word} which
 looks upright.}
```

In this emphasized sentence, there is an emphasized word*which looks upright.*

Text may be emphasized more heavily through the use of boldface, particularly for keywords the reader may be trying to find when reading the text. As bold text is generally read before any other text in a paragraph or even on a page, it should be used sparingly. It may also be used in place of italics when using sans-serif typefaces to provide a greater contrast with unemphasized text. Bold text can be generated with the `\textbf{text}` command.

```
\textbf{Bold text} may be used to heavily emphasize very important words or
 phrases.
```

Bold text may be used to heavily emphasize very important words or phrases.

9.5. Font encoding

A *character* is a sequence of bytes, and should not be confused with its representation, the *glyph* , which is what the reader sees. So the character 'a' has different representations following the used font, for example the upright version, the italic version, various weights and heights, and so on.

Upon compilation, `tex` will have to choose the right font glyph for every character. This is what is called *font encoding* . The default LaTeX font encoding is OT1, the encoding of the original Computer Modern TeX text fonts. It contains only 128 characters, many from ASCII, but leaving out some others and including a number that are not in ASCII. When accented characters are required, TeX creates them by combining a normal character with an accent. While the resulting output looks perfect, this approach has some caveats.

- It stops the automatic hyphenation from working inside words containing accented characters.
- Searches for words with accents in PDFs will fail.

- Extracting ('e.g.' copy paste) the umlaut 'Ä' via a PDF viewer actually extracts the two characters '`"A`'.
- Besides, some of Latin letters could not be created by combining a normal character with an accent, to say nothing about letters of non-Latin alphabets, such as Greek or Cyrillic.

To overcome these shortcomings, several 8-bit CM-like font sets were created. *Extended Cork* (EC) fonts in T1 encoding contains letters and punctuation characters for *most of the European languages* based on Latin script. The LH font set contains letters necessary to typeset documents in languages using Cyrillic script. Because of the large number of Cyrillic glyphs, they are arranged into four font encodings—T2A, T2B, T2C, and X2. The CB bundle contains fonts in LGR encoding for the composition of Greek text. By using these fonts you can improve/enable hyphenation in non-English documents. Another advantage of using new CM-like fonts is that they provide fonts of CM families in all weights, shapes, and optically scaled font sizes.

All this is not possible with *OT1* ; that's why you may want to change the font encoding of your document.

⚠ **Warning**

If you do not have a specific font encoding issue (*e.g.* writing English only), there is no need for T1. Sticking to the default font encoding is not a problem.

Note that changing the font encoding will have some requirements over the fonts being used. The default Computer Modern font does not support T1. You will need Computer Modern Super (`cm-super`) or Latin Modern (`lmodern`), which are Computer Modern-like fonts with T1 support. If you have none of these, it is quite frequent (depends on your TeX installation) that `tex` chooses a Type3 font such as the Type3 EC, which is a bitmap font. Bitmap fonts look rather ugly when zoomed or printed.

⚠ **Warning**

If after using T1 you find yourself with very low quality fonts, it is because there is no appropriate font installed on your system. Install either `cm-super` or `lmodern`. This is a very common error!

The `fontenc` package tells LaTeX what font encoding to use. Font encoding is set with:

```
\usepackage['encoding']{fontenc}
```

where `encoding` is the font encoding. It is possible to load several encodings simultaneously.

There is nothing to change in your document to use CM Super fonts (assuming they are installed), they will get loaded automatically if you use T1 encoding. For `lmodern` , you will need to load the package after the T1 encoding has been set:

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

The package `ae` (almost European) is obsolete. It provided some workarounds for hyphenation of words with special characters. These are not necessary any more with fonts like lmodern. Using the ae package leads to text encoding problems in PDF files generated via `pdflatex` (e.g. text extraction and searching), besides typographic issues.

9.6. Font styles

Each family has its own font characteristics (such as italic and bold), also known as font styles, or font properties.

Font styles are usually implemented with different font files. So it is possible to build a new font family by specifying the font styles of different font families.

9.6.1. Shapes

The following table lists the commands you will need to access the typical font shapes:

LaTeX command	**Equivalent to**	**Output style**	**Remarks**
`\textnormal{...}`	`{\normalfont ...}`	document font family	This is the default or normal font.
`\emph{...}`	`{\em ...}`	*emphasis*	Typically italics. Using emph{} inside of italic text removes the italics on the emphasized text.
`\textrm{...}`	`{\rmfamily ...}`	roman font family	
`\textsf{...}`	`{\sffamily ...}`	sans serif font family	
`\texttt{...}`	`{\ttfamily ...}`	teletypefont family	This is a fixed-width or monospace font.
`\textup{...}`	`{\upshape ...}`	upright shape	The same as the normal typeface.
`\textit{...}`	`{\itshape ...}`	italic shape	
`\textsl{...}`	`{\slshape ...}`	slanted shape	A skewed version of the normal typeface (similar to, but slightly different from, italics).
`\textsc{...}`	`{\scshape ...}`	Small Capitals	
`\uppercase{...}`		uppercase (all caps)	Also `\lowercase`. There are some caveats, though; see here[11].
`\textbf{...}`	`{\bfseries ...}`	bold	
`\textmd{...}`	`{\mdseries ...}`	medium weight	A font weight in between normal and bold.

11 `http://www.tex.ac.uk/cgi-bin/texfaq2html?label=casechange`

LaTeX command	Equivalent to	Output style	Remarks
`\textlf{...}`	`{\lfseries ...}`	light	A font weight lighter than normal. Not supported by all typefaces.

The commands in column two are not entirely equivalent to the commands in column one: They do not correct spacing after the selected font style has ended. The commands in column one are therefore in general recommended.

You may have noticed the absence of underline. This is because underlining is not recommended for typographic reasons (it weighs the text down). You should use *emph* instead. However underlining text provides a useful extra form of emphasis during the editing process, for example to draw attention to changes. Although underlining is available via the `\underline{...}` command, text underlined in this way will not break properly. This functionality has to be added with the `ulem` (underline emphasis) package. Stick `\usepackage{ulem}` in your preamble. By default, this overrides the `\emph` command with the underline rather than the italic style. It is unlikely that you wish this to be the desired effect, so it is better to stop `ulem` taking over `\emph` and simply call the underline command as and when it is needed.

- To restore the usual `\emph` formatting, add `\normalem` straight after the `document` environment begins. Alternatively, use `\usepackage[normalem]{ulem}`.
- To underline, use `\uline{...}`.
- To add a wavy underline, use `\uwave{...}`.
- For a strike-out (strikethrough), use `\sout{...}`.
- For a slash through each individual character `\xout{...}`.

Some font styles are not compatible one with the other. But some extra packages will fill this hole. For bold small capitals, you might want to use:

```
\usepackage{bold-extra}
% ...
\textsc{ \textbf{This is bold small capitals} }
```

9.6.2. Sizing text

To apply different font sizes, simply follow the commands on this table:

Command	Output
`\tiny`	sample text
`\scriptsize`	sample text
`\footnotesize`	sample text
`\small`	sample text
`\normalsize`	sample text
`\large`	sample text
`\Large`	sample text
`\LARGE`	sample text
`\huge`	sample text
`\Huge`	sample text

These commands change the size within a given scope, so for instance `{\Large some words}` will change the size of only `some words`, and does not affect the font in the rest of the document. It will work for most parts of the text.

```
{\Large\tableofcontents}
```

These commands cannot be used in math mode. However, part of a formula may be set in a different size by using an \mbox command containing the size command. The new size takes effect immediately after the size command; if an entire paragraph or unit is set in a certain size, the size command should include the blank line or the `\end{...}` which delimits the unit.

The default for `\normalsize` is 10 point (option `10pt`), but it may differ for some Document Styles or their options. The actual size produced by these commands also depends on the Document Style and, in some styles, more than one of these size commands may produce the same actual size.

Note that the font size definitions are set by the document class. Depending on the document style the actual font size may differ from that listed above. And not every document class has unique sizes for all 10 size commands.

Note: the following table is mostly wrong. Until someone gets around to fixing it: use \makeatletter and \f@size to find out the font size.

Absolute Point Sizes, [10pt] being default										
size	**standard classes,** *proc*			AMS classes, *memoir*			*slides*	*beamer*		
	[10pt]	**[11pt]**	**[12pt]**	**[10pt]**	**[11pt]**	**[12pt]**		**[10pt]**	**[11pt]**	**[12pt]**
`\tiny`	6.80565	7.33325	7.33325	7.33325	7.97224	8.50012	17.27505	5.31258	6.37509	6.37509
`\scriptsize`	7.97224	8.50012	8.50012	7.97224	8.50012	9.24994	20.73755	7.43760	8.50012	8.50012
`\footnotesize`	8.50012	9.24994	10.00002	8.50012	9.24994	10.00002	20.73755	8.50012	9.24994	10.00002
`\small`	9.24994	10.00002	10.95003	9.24994	10.00002	10.95003	20.73755	9.24994	10.00002	10.95003
`\normalsize`	10.00002	10.95003	11.74988	10.00002	10.95003	11.74988	24.88382	10.00002	10.95003	11.74988
`\large`	11.74988	11.74988	14.09984	10.95003	11.74988	14.09984	29.86258	11.74988	11.74988	14.09984
`\Large`	14.09984	14.09984	15.84985	11.74988	14.09984	15.84985	35.82510	14.09984	14.09984	16.24988
`\LARGE`	15.84985	15.84985	19.02350	14.09984	15.84985	19.02350	43.00012	16.24988	16.24988	19.50362
`\huge`	19.02350	19.02350	22.82086	15.84985	19.02350	22.82086	51.60014	19.50362	19.50362	23.39682
`\Huge`	22.82086	22.82086	22.82086	19.02350	22.82086	22.82086	51.60014	23.39682	23.39682	23.39682

As a technical note, points in TeX follow the standard American point size in which 1 pt is approximately 0.3513 6-1pt¯ mm. The standard point size used in most modern computer programs (known as the *desktop publishing point* or *PostScript point*) has 1 pt equal to approximately 0.352 7-1pt¯ mm while the standard European point size (known as the *Didot point*) had 1 pt equal to approximately 0.37597151 mm (see: point (typography)[12]).

9.7. Local font selection

You can change font for a specific part of the text. There are four font properties you can change.

`\fontencoding`

The font encoding, such as OT1 (TeX default) or T1 (extended characters support, better PDF support, widely used).

`\fontfamily`

The font family.

`\fontseries`

The series: l=light, m=medium, b=bold, bx=very bold.

`\fontshape`

The shape: it=italic, n=normal, sl=slanted, sc=small capitals.

```
{
\fontfamily{anttlc}\selectfont
Some text in anttlc...
}
```

The `\selectfont` command is mandatory, otherwise the font will not be changed. It is highly recommended to enclose the command in a group to cleanly return to the previous font selection when desired.

You can use all these commands in a row:

```
{
\fontencoding{T1}\fontfamily{anttlc}\fontseries{m}\fontshape{n}\selectfont
Some text in anttlc...
}
```

The default values are stored in `\encodingdefault`, `\familydefault`, `\seriesdefault` and `\shapedefault`. Setting back the default font properties can be done with

```
\fontencoding{\encodingdefault}
\fontfamily{\familydefault}
\fontseries{\seriesdefault}
\fontshape{\shapedefault}
\selectfont
```

For short, you can use the `\usefont{<encoding>}{<family>}{<series>}{<shape>}` command.

12 `http://en.wikipedia.org/wiki/Point_%28typography%29`

```
\usefont{T1}{cmr}{m}{n} % Computer Modern Roman (TeX default) in T1 encoding.
 May lead to bad text quality if you do not have cm-super installed.
\usefont{T1}{phv}{m}{sc} % phv family (sans serif) medium small capitals.
\usefont{T1}{ptm}{b}{it} % ptm family bold italic
\usefont{U}{pzd}{m}{n}   % ...
```

9.8. Arbitrary font size

The `\tiny`...`\Huge` commands are often enough for most contents. These are fixed sizes however. In most document processors, you can usually choose any size for any font. This is because the characters actually get magnified. If it usually looks correct for medium sizes, it will look odd at extreme sizes because of an unbalanced thickness. In TeX it is possible to change the magnification of anything, but highly discouraged for the aforementioned reason. Changing the font size is made by changing the font file. Yes, there is a file for every size: cmr10 for Computer Modern Roman 10pt, cmr12 for Computer Modern Roman 12pt, etc. This ensure the characters are correctly balanced and remain readable at all defined sizes.

You may choose a particular font size with the `\fontsize{<size>}{<line space>}` command. Example:

```
{\fontsize{5cm}{1em}\selectfont This is big!}
```

If you are using the default Computer Modern font with OT1 encoding, you may get the following message:

```
LaTeX Font Warning: Font shape `OT1/cmr/m/n' in size <142.26378> not available
(Font)              size <24.88> substituted on input line 103.
```

In that case you will notice that the font size cannot be changed beyond `\tiny` and `\Huge`. You must switch to a more sizable font, *e.g.*

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

9.9. Finding fonts

You will find a huge font directory along examples and configurations at TUG Font Catalogue[13].

13 `http://www.tug.dk/FontCatalogue/`

9.10. Using arbitrary system fonts

If you use the XeTeX[14] or LuaTeX[15] engine and the fontspec[16] package, you'll be able to use any font installed in the system effortlessly. XeTeX also allows using OpenType[17] technology of modern fonts like specifying alternate glyphs and optical size variants. XeTeX also uses Unicode[18] by default, which might be helpful for font issues.

To use the fonts, simply load the `fontspec` package and set the font:

```
\documentclass{article}

\usepackage{fontspec}
\setmainfont{Arial}

\begin{document}
Lorem ipsum...
\end{document}
```

Then compile the document with `xelatex` or `lualatex` . Note that you can only generate .pdf files, and that you need a sufficiently new TeX distribution (TeX Live 2009 should work for XeTeX and Tex Live 2010 for LuaTeX). Also you should *not* load the `inputenc` or `fontenc` package. Instead make sure that your document is encoded as UTF-8 and load `fontspec`, which will take care of the font encoding. To make your document support both `pdflatex` and `xelatex` /`lualatex` you can use the `\ifxetex`/ `\ifluatex` macro from the ifxetex[19]/ ifluatex[20] package. For example for `xelatex`

```
\documentclass{article}
\usepackage{ifxetex}

\ifxetex
  \usepackage{fontspec}
  \defaultfontfeatures{Ligatures=TeX} % To support LaTeX quoting style
  \setromanfont{Hoefler Text}
\else
  \usepackage[T1]{fontenc}
  \usepackage[utf8]{inputenc}
\fi

\begin{document}
Lorem ipsum...
\end{document}
```

9.11. PDF fonts and properties

PDF documents have the capability to embed font files. It makes them portable, hence the name *Portable Document Format* .

14 `http://en.wikipedia.org/wiki/XeTeX`
15 `http://en.wikipedia.org/wiki/LuaTeX`
16 `http://www.ctan.org/pkg/fontspec`
17 `http://en.wikipedia.org/wiki/Opentype`
18 `http://en.wikipedia.org/wiki/Unicode`
19 `http://www.ctan.org/pkg/ifxetex/`
20 `http://www.ctan.org/pkg/ifluatex`

Many PDF viewers have a *Properties* feature to list embedded fonts and document metadata.

Many Unix systems make use of the *poppler* tool set which features `pdfinfo` to list PDF metadata, and `pdffonts` to list embedded fonts.

9.12. Useful websites

- The Latex Font Catalogue[21]
- LaTeX font commands[22]
- How to change fonts in Latex[23]
- [ftp://tug.ctan.org/tex-archive/fonts/utilities/fontinst/doc/talks/et99-font-tutorial.pdf Understanding the world of TEX fonts and mastering the basics of fontinst]
- Font installation the shallow way[24] *"For one-off projects, you can cut corners with font installation* (i.e. fontinst) *and end up with a more manageable set of files and a cleaner TEX installation. This article shows how and why"*

9.12.1. TrueType (ttf) fonts

- Step-by-step guide to manually install a ttf-font for PdfTeX[25]
- A bash script for installing a LaTeX font family[26] (MikTeX[27] / TeXLive[28])
- LaTeX And TrueType Font[29]
- True Type Fonts with LaTeX under Linux + MiKTeX 2.5[30]
- Unicode Truetype font installer for LaTeX under Windows + MikTeX[31]
- Using TrueType fonts with TeX (LaTeX) and pdfTeX (pdfLaTeX)[32] (for MikTeX)

9.13. References

21 `http://www.tug.dk/FontCatalogue/`
22 `http://www.cl.cam.ac.uk/~rf10/pstex/latexcommands.htm`
23 `http://www.ee.iitb.ac.in/~trivedi/LatexHelp/latexfont.htm`
24 `http://www.tug.org/TUGboat/Articles/tb27-1/tb86kroonenberg-fonts.pdf`
25 `http://c.caignaert.free.fr/Install-ttf-Font.pdf`
26 `http://www.tex.ac.uk/ctan/support/installfont/`
27 `http://latex.josef-kleber.de/download/installfont-tl`
28 `http://latex.josef-kleber.de/download/installfont-tl`
29 `http://xpt.sourceforge.net/techdocs/language/latex/latex33-LaTeXAndTrueTypeFont`
30 `http://fachschaft.physik.uni-greifswald.de/~stitch/ttf.html`
31 `http://william.famille-blum.org/software/latexttf/index.html`
32 `http://www.radamir.com/tex/ttf-tex.htm`

10. List Structures

Convenient and predictable list formatting is one of the many advantages of using LaTeX. Users of WYSIWYG word processors can sometimes be frustrated by the software's attempts to determine when they intend lists to begin and end. As a mark-up language, LaTeX gives more control over the structure and content of lists.

10.1. List structures

Lists often appear in documents, especially academic, as their purpose is often to present information in a clear and concise fashion. List structures in LaTeX are simply environments which essentially come in three flavors: `itemize`, `enumerate` and `description`.

All lists follow the basic format:

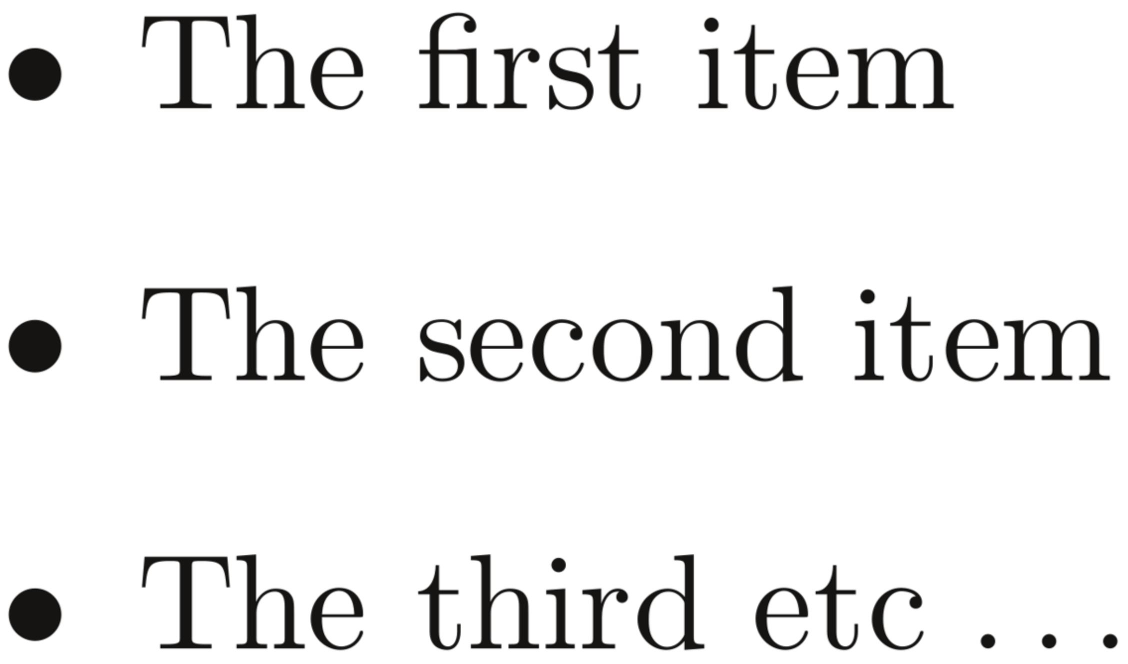
```
\begin{list_type}

  \item The first item
  \item The second item
  \item The third etc \ldots

\end{list_type}
```

All three of these types of lists can have multiple paragraphs per item: just type the additional paragraphs in the normal way, with a blank line between each. So long as they are still contained within the enclosing environment, they will automatically be indented to follow underneath their item.

10.1.1. Itemize

This environment is for your standard bulleted list of items.

```
\begin{itemize}
  \item The first item
  \item The second item
  \item The third etc \ldots
\end{itemize}
```



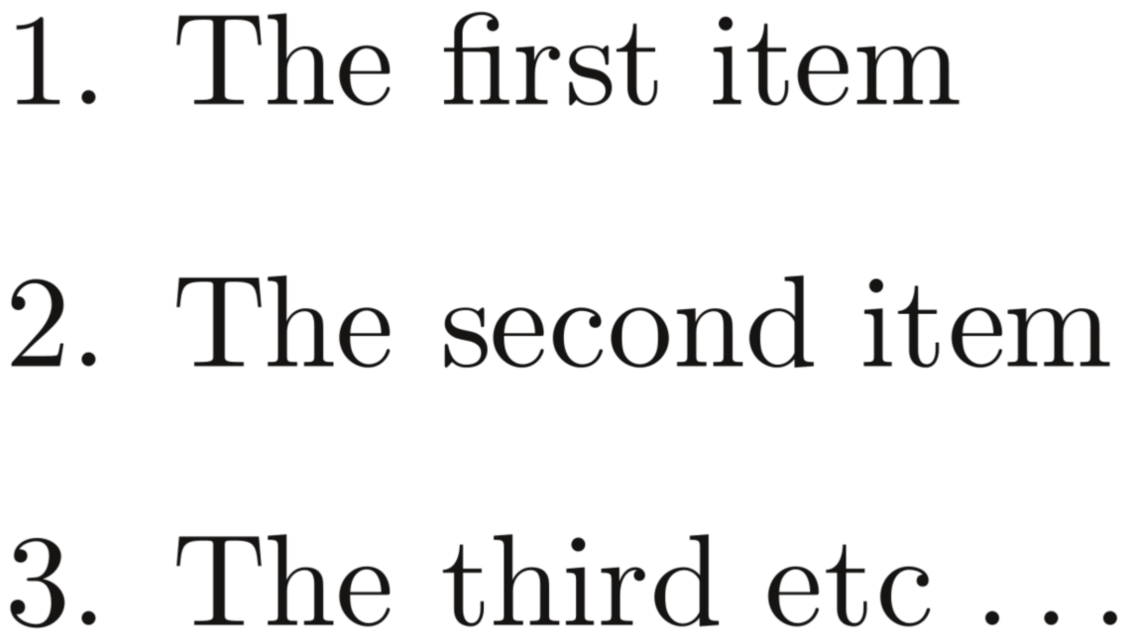
- The first item
- The second item
- The third etc …

Figure 28

10.1.2. Enumerate

The enumerate environment is for ordered lists, where by default, each item is numbered sequentially.

```
\begin{enumerate}
  \item The first item
  \item The second item
  \item The third etc \ldots
\end{enumerate}
```



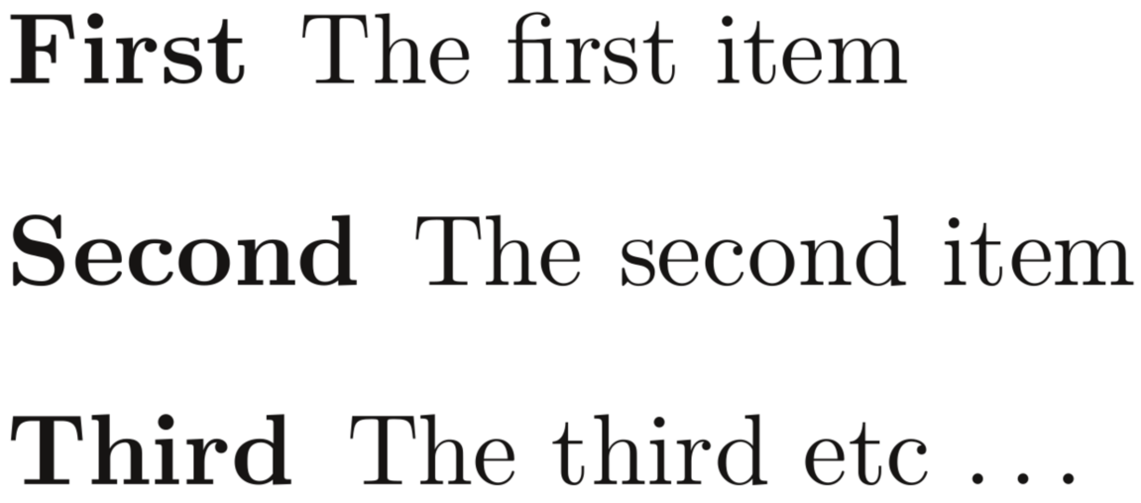
1. The first item
2. The second item
3. The third etc . . .

Figure 29

10.1.3. `Description`

The description environment is slightly different. You can specify the item label by passing it as an optional argument (although optional, it would look odd if you didn't include it!). Ideal for a series of definitions, such as a glossary.

```
\begin{description}
  \item[First] The first item
  \item[Second] The second item
  \item[Third] The third etc \ldots
\end{description}
```



First The first item

Second The second item

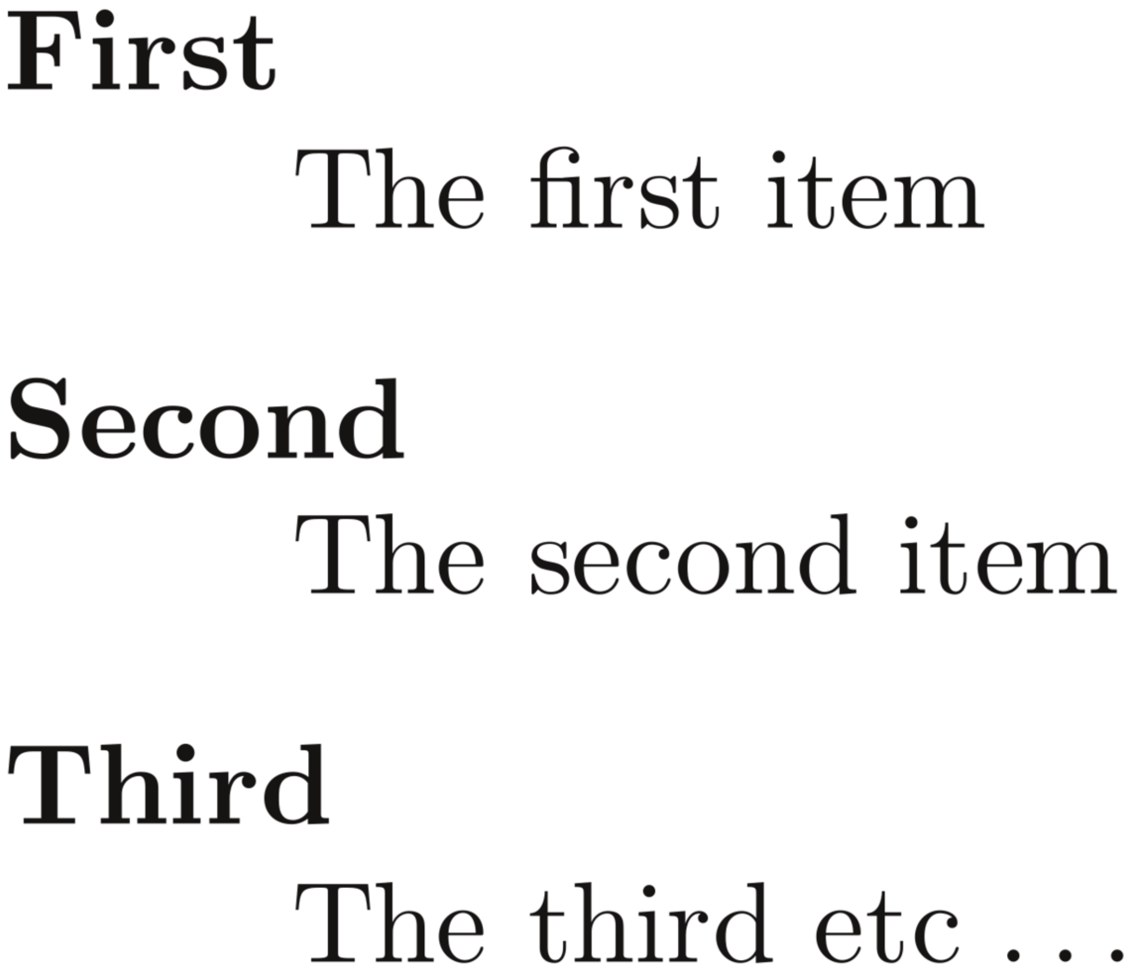
Third The third etc ...

Figure 30

Sometimes you want a description where the text begins on a new line. This cannot easily be done with `\\`. The trick is to use `\hfill`[1].

```
\begin{description}
  \item[First] \hfill \\
  The first item
  \item[Second] \hfill \\
  The second item
  \item[Third] \hfill \\
  The third etc \ldots
\end{description}
```

1 `http://www.tex.ac.uk/cgi-bin/texfaq2html?label=noline`



First

The first item

Second

The second item

Third

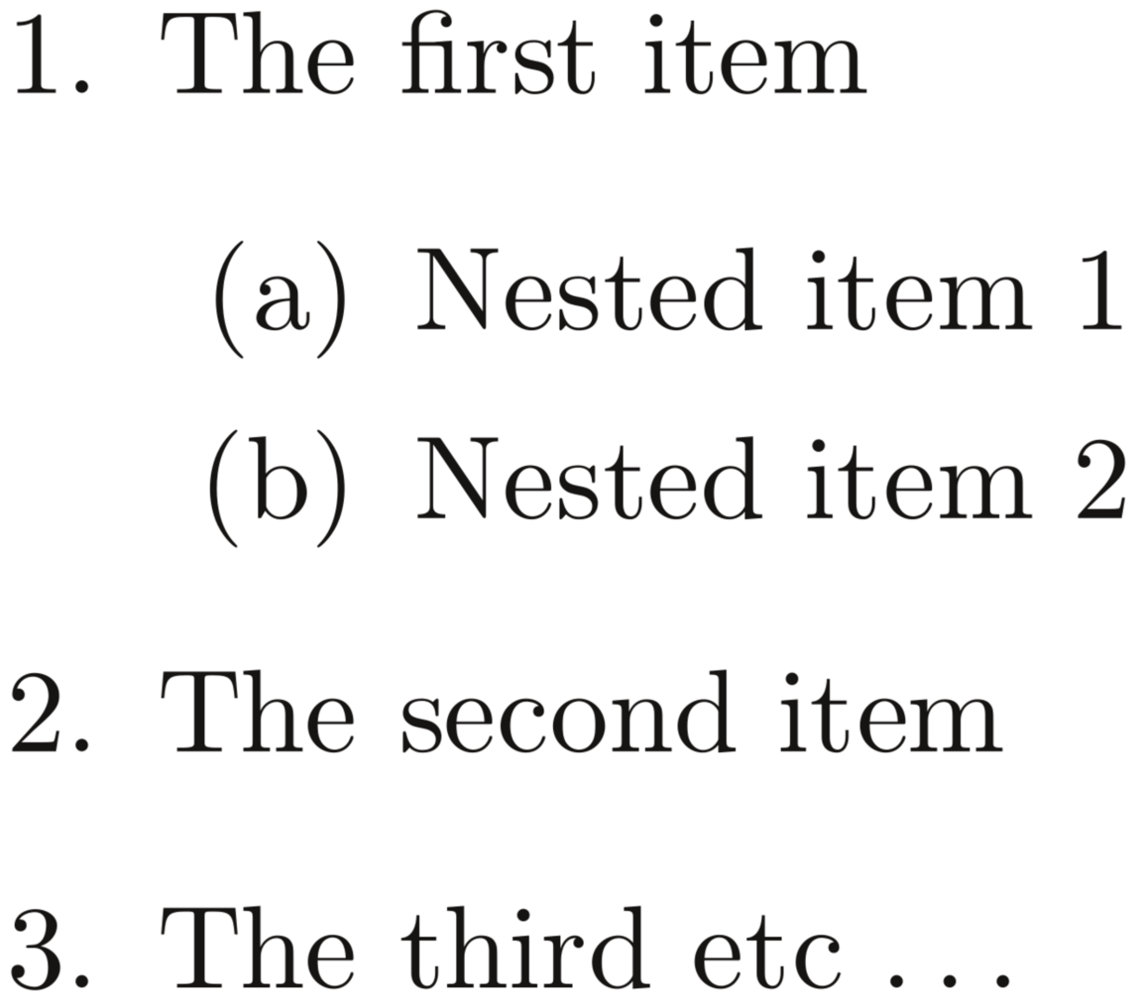
The third etc . . .

Figure 31

10.2. Nested lists

LaTeX will happily allow you to insert a list environment into an existing one (up to a depth of four—if you need more than four, use the `easylist` package). Simply begin the appropriate environment at the desired point within the current list. Latex will sort out the layout and any numbering for you.

```
\begin{enumerate}
  \item The first item
  \begin{enumerate}
    \item Nested item 1
    \item Nested item 2
  \end{enumerate}
  \item The second item
  \item The third etc \ldots
\end{enumerate}
```



1. The first item
   (a) Nested item 1
   (b) Nested item 2
2. The second item
3. The third etc . . .

Figure 32

10.3. Customizing lists

Customizing LaTeX is outside the beginners' domain. While not necessarily difficult in itself, because beginners are already overwhelmed with the array of commands and environments, moving on to more advanced topics runs the risk of confusion.

However, since the tutorial aims at being complete, we shall still include a brief guide on customizing lists. Feel free to skip!

Note that in the following when `\renewcommand` is used it must appear after the `\begin{document}` instruction so the changes made are taken into account. This is needed for both enumerated and itemized lists.

Also beware of the spaces in the label definitions. It is a common error!

10.3.1. Line spacing

As you may have noticed, in standard LaTeX document classes, the vertical spacing between items, and above and below the lists as a whole, is more than between paragraphs: it may look odd if the descriptions are too short.

Using packages

If you want tightly-packed lists, use the `mdwlist` package (included in the `mdwtools` bundle), which provides compact, "starred" versions of the previous environments, i.e. `itemize*`, `enumerate*` and `description*`. They work exactly in the same way, but the output is more compact. Other packages providing compacted lists are `paralist` and `enumitem`.

Alternatively, use the `memoir` class and with `\tightlists`.

Customizing manually

Inside lists you can redefine some length/dimension variables of LaTeX, for example using:

```
\begin{itemize} \itemsep1pt \parskip0pt \parsep0pt
  \item first item
  \item second item
\end{itemize}
```

Alternatively, to create a unified look in your document you can redefine the enumerate environment:

```
\let\oldenumerate\enumerate
\renewcommand{\enumerate}{
  \oldenumerate
  \setlength{\itemsep}{1pt}
  \setlength{\parskip}{0pt}
  \setlength{\parsep}{0pt}
}
```

Another approach is to redefine the `\item` command globally.

```
\newlength{\wideitemsep}
\setlength{\wideitemsep}{.5\itemsep}
\addtolength{\wideitemsep}{-7pt}
\let\olditem\item
\renewcommand{\item}{\setlength{\itemsep}{\wideitemsep}\olditem}
```

10.3.2. Customizing enumerated lists

Using packages

The thing people want to change most often with Enumerated lists are the counters. A quick solution to this problem is provided by the `enumerate` package of David Carlisle[2], or

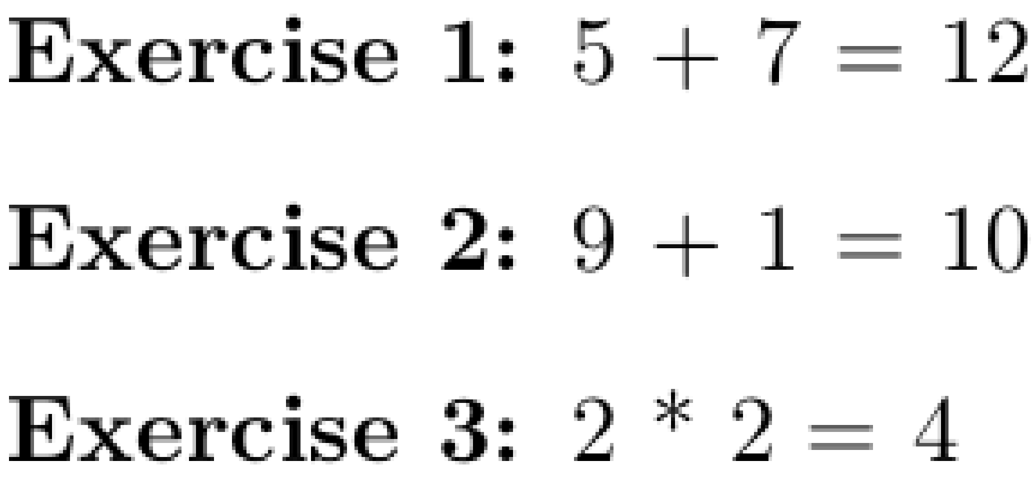
2 `http://mirrors.fe.up.pt/pub/CTAN/macros/latex/required/tools/enumerate.pdf`The enumerate package, David Carlisle 1999

the more sophisticated package `enumitem` by Javier Bezos[3]. When using `enumerate`, it is possible to specify the *style* of the numbering: `\begin{enumerate}[style]`.

The options *A* , *a* , *I* , *i* and *1* define the style and are self-explanatory, anything else is treated as text. To use any of the style tokens as text they can be enclosed in braces, e.g. `{A}` will give a literal A. <ref ftp://tug.ctan.org/ctan/macros/latex/required/tools/enumerate.pdf> CTAN documentation for `enumerate`

Sometimes you may want to place some short text in front of the enumeration for example: "Exercise 1, Exercise 2, Execise 3, ...". This is possible with the `enumitem` package:

```
\begin{enumerate}[label=\bfseries Exercise \arabic*:]
  \item 5 + 7 = 12
  \item 9 + 1 = 10
  \item 2 * 2 = 4
\end{enumerate}
```


Exercise 1: 5 + 7 = 12

Exercise 2: 9 + 1 = 10

Exercise 3: 2 * 2 = 4

Figure 33 Enumeration with text

\bfseries makes it bold, *Exercise* is the text and *\arabic** inserts the counter followed by a *colon (:)* which is treated as text again.

3 `http://mirrors.fe.up.pt/pub/CTAN/macros/latex/contrib/enumitem/enumitem.pdf`The enumitem package, Javier Bezos 2011

Manually

To go any further and do it yourself instead, a brief introduction to LaTeX *counters* is required. You should check the dedicated chapter[4] as we will not delve into the details for now.

There are four individual counters that are associated with itemized lists, each one represents the four possible levels of nesting, which are called:

```
enumi
enumii
enumiii
enumiv
```

The counter is incremented by `\item` before it is printed. For example to reset `enumi` use:

```
\begin{enumerate}
  \setcounter{enumi}{4}
  \item fifth element
\end{enumerate}
```

5. fifth element

The command responsible for formatting the various levels of nesting are

```
\labelenumi
\labelenumii
\labelenumiii
\labelenumiv
```

Example:

```
\renewcommand{\labelenumi}{(\Roman{enumi})}
\renewcommand{\labelenumii}{\Roman{enumi}.~\alph{enumii}}
```

This simply redefines the appearance of the label, which is fine, providing that you do not intend to cross-reference to a specific item within the list, in which case the reference will be printed in the original format. This issue does not arise if you redefine the counter printer:

```
\renewcommand{\theenumi}{\Roman{enumi}}
\renewcommand{\labelenumi}{(\theenumi)}
\renewcommand{\theenumii}{\alph{enumii}}
\renewcommand{\labelenumii}{\theenumi.~\theenumii}
```

10.3.3. Customizing itemized lists

Itemized lists are not as complex as they do not need to count. Therefore, to customize, you simply change the labels. It can be done manually for each entry with `\item[new symbol]`, eg `\item[$\star$]`.

4 Chapter 24 on page 291

The itemize labels are accessed via `\labelitemi`, `\labelitemii`, `\labelitemiii`, `\labelitemiv`, for the four respective levels.

```
\renewcommand{\labelitemi}{\textgreater}
```

The above example would set the labels for the first level to a greater than (>) symbol. Of course, the text symbols available in Latex are not very exciting. Why not use one of the ZapfDingbat symbols, as described in the Symbols[5] section. Or use a mathematical symbol:

```
\renewcommand{\labelitemi}{$\star$}
```

Itemized list with tightly set items, that is with no vertical space between two consecutive items, can be created as follows.

```
\begin{itemize}
  \setlength{\itemsep}{0cm}%
  \setlength{\parskip}{0cm}%
  \item Item opening the list
  \item Item tightly following
\end{itemize}
```

10.4. Inline lists

Inline lists can be achieved as follows.

10.4.1. With package paralist

The `paralist` package provides the `inparaenum` environment (with an optional formatting specification in square brackets):

```
\usepackage{paralist}
% ...

\begin{document}

Inline lists, which are
sequential in nature, just like enumerated
lists, but are
\begin{inparaenum}[\itshape a\upshape)]
\item formatted within their paragraph;
\item usually labelled with letters; and
\item usually have the final item prefixed with
`and' or `or',
\end{inparaenum} like this example.
...
```

5 Chapter 10.4.2 on page 119

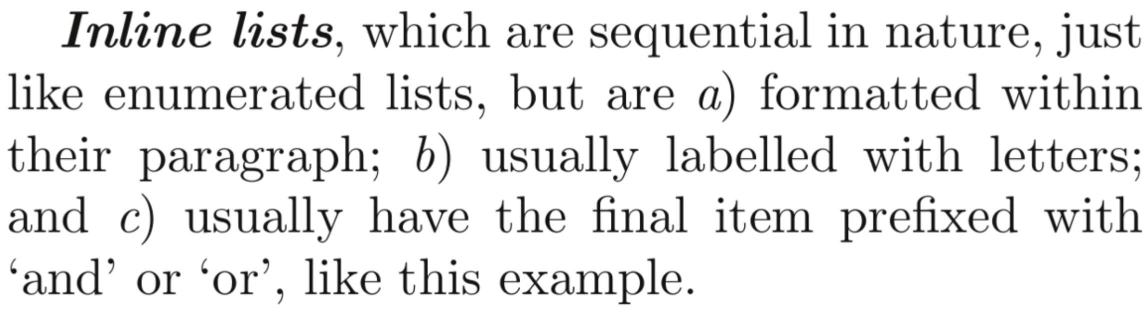

Inline lists, which are sequential in nature, just like enumerated lists, but are *a*) formatted within their paragraph; *b*) usually labelled with letters; and *c*) usually have the final item prefixed with ‘and’ or ‘or’, like this example.

Figure 34

To change the styles of the counter, tokens A, a, I, i, and 1 can be used in the optional argument to produce the counter with one of the styles `\Alph`, `\alph`, `\Roman`, `\roman` and `\arabic`. For example: `\begin{inparaenum}[(i)]` produces the labels (i), (ii), (iii) ...

10.4.2. With package `enumitem`

```
\usepackage[inline]{enumitem}
% ...

\begin{document}

Inline lists, which are
sequential in nature, just like enumerated
lists, but are
\begin{enumerate*}[label=\itshape\alph*\upshape)]
\item formatted within their paragraph;
\item usually labelled with letters; and
\item usually have the final item prefixed with
`and' or `or',
\end{enumerate*} like this example.
...
```

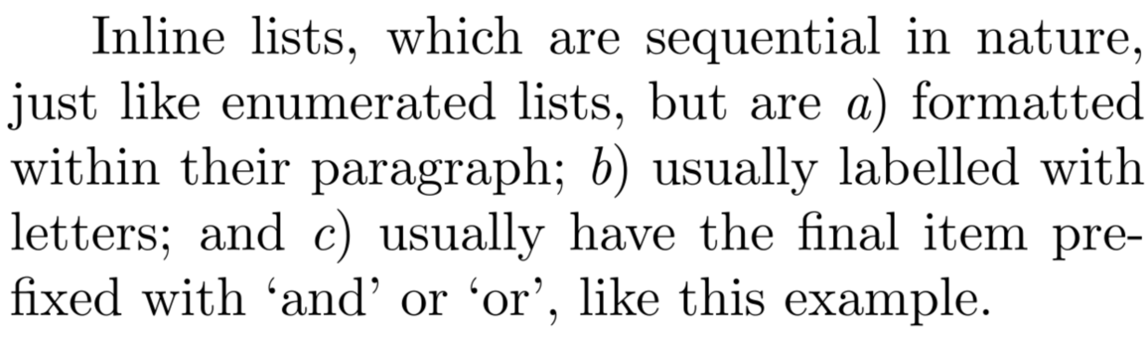

Inline lists, which are sequential in nature, just like enumerated lists, but are *a*) formatted within their paragraph; *b*) usually labelled with letters; and *c*) usually have the final item prefixed with ‘and’ or ‘or’, like this example.

Figure 35

Package `shortlst` also provides inline lists.

10.5. Easylist package

The `easylist` package <ref ftp://ctan.tug.org/tex-archive/macros/latex/contrib/easylist/easylist-doc.pdf>easylist documentation allows you to create list using a more convenient syntax and with infinite nested levels. It is also very customizable.

Load the package with the control character as optional argument:

```
\usepackage[ampersand]{easylist}
```

The `easylist` environment will default to enumerations.

```
\begin{easylist}
& Main item~:
&& Sub item.
&& Another sub item.
\end{easylist}
```

It features predefined styles which you can set as optional argument.

```
\begin{easylist}[itemize]
% ...
\end{easylist}
```

Available styles:

- `tractatus`
- `checklist` - All items have empty check boxes next to them
- `booktoc` - Approximately the format used by the table of contents of the book class
- `articletoc` - Approximately the format used by the table of contents of the article class
- `enumerate` - The default
- `itemize`

You can customize lists with the `\ListProperties(...)` command and revert back the customization with `\newlist{}`. Yes, that's parentheses for `\ListProperties` parameters.

The `Style` parameter sets the style of counters and text, the `Style*` parameter sets the sytle of counters, and the `Style**` parameter sets the style of text. The parameter `Numbers` determines the way that the numbers are displayed and the possible values are `r` or `R` (for lower and upper case Roman numerals), `l` or `L` (for lower and upper case letters), `a` (for Arabic numbers, the default), and `z` (for Zapf's Dingbats).

The `FinalMark` parameter sets the punctuation of the final counter (Ex: `FinalMark3={)}`) while `FinalSpace` sets the amount of space between the item and the item's text. The `Margin` parameter sets the distance from the left margin (Ex: `FinalSpace2=1cm`). The `Progressive` parameter sets the distance from the left margin of all items in proportion to their level.

The parameter prevents the first `n` counters from appearing in all levels. If there is a number after a parameter (Ex: `Style3*`) then this numbers indicates the level that it will affect (Ex: `Style3=\color{red}`).

Example of custom enumerate:

```
\begin{easylist}[enumerate]
\ListProperties(Style2*=,Numbers=a,Numbers1=R,FinalMark={)})
```

```
& Main item~:
&& Sub item.
&& Another sub item.
\end{easylist}
```

Note that we put the `FinalMark` argument between `{}` to avoid LaTeX understanding it as the end of the properties list. Now we change the default properties to print a custom itemize:

```
\usepackage{amssymb}
\ListProperties(Hide=100, Hang=true, Progressive=3ex, Style*=-- ,
Style2*=$\bullet$ ,Style3*=$\circ$ ,Style4*=\tiny$\blacksquare$ )
% ...

\begin{easylist}
& Blah
& Blah
&& Blah
&&& Blah
&&&& Blah
&&&&& Blah
\end{easylist}
```

– Blah

  • Blah

    ∘ Blah

      ■ Blah

        – Blah

Spaces in `Style` parameters are important. The `Style*` parameter acts as a default value and `easylist` will use a medium dash for level 1, 5 and onward.

You can also define custom styles using LaTeX macros:

```
\newcommand\myitemize{\ListProperties(Hide=100, Hang=true, Progressive=3ex,
 Style*=$\star$ )}
\newcommand\myenumerate{\ListProperties(Space=2\baselineskip)}

% ...
\begin{easylist} \myitemize
& Blah
\end{easylist}
```

Important note: `easylist` has some drawbacks. First if you need to put an easylist inside an environment using the same control character as the one specified for easylist, you may get an error. To circumvent it, use the following commands provided by easylist:

```
\Activate
\begin{easylist}
& ...
```

```
\end{easylist}
\Deactivate
```

Besides using `easylist` along with figures may cause some trouble to the layout and the indentation. LaTeX lists do not have this problem.

To use easylist with Beamer, each frame that uses easylist must be marked as fragile:

```
\begin{frame}[fragile]
    ...
    \begin{easylist}[itemize]
        ...
    \end{easylist}
    ...
\end{frame}
```

10.6. Notes and references

11. Special Characters

In this chapter we will tackle matters related to input encoding, typesetting diacritics and special characters.

In the following document, we will refer to *special characters* for all symbols other than *A-Za-z0-9* and English punctuation marks.

This chapter is tightly linked with the font encoding issue. You should have a look at Fonts[1] on the topic.

Some languages usually need a dedicated input system to ease document writing. This is the case for Arabic, Chinese, Japanese, Korean and others. This specific matter will be tackled in Internationalization[2].

The rules for producing characters with diacritical marks, such as accents, differ somewhat depending whether you are in text mode, math mode, or the tabbing environment.

11.1. Input encoding

11.1.1. A technical matter

Most of the modern computer systems allow you to input letters of national alphabets directly from the keyboard. If you tried to input these special characters in your LaTeX source file and compiled it, you may have noticed that they do not get printed at all.

A LaTeX source document is a plain text file. A computer stores data in a binary format, that is a sequence of bits (0 and 1). To display a plain text file, we need a code which tells which sequence of bits corresponds to which sequence of characters. This association is called *input encoding* , *character encoding* , or more informally *charset* .

For historical reasons, there are many different input encodings. There is an attempt to unify all the encoding with a specification that contains all existent symbols that are known from human history. This specification is *Unicode* . It only defines *code points* , which is a number for a symbol, but not the way symbols are represented in binary value. For that, unicode encodings are in charge. There are also several unicode encodings available, UTF-8 being one of them.

The ASCII encoding is an encoding which defines 128 characters on 7 bits. Its widespread use has led the vast majority of encodings to have backward compatibility with ASCII, by

1 Chapter 9 on page 95
2 Chapter 12 on page 133

defining the first 128 characters the same way. The other characters are added using more bits (8 or more).

This is actually a big issue, since if you do not use the right encoding to display a file, it will show weird characters. What most programs try to do is guess statistically the encoding by analyzing the frequent sequences of bits. Sadly, it is not 100% safe. Some text editors may not bother guessing the encoding and will just use the OS default encoding. You should consider that other people might not be able to display directly your input files on their computer, because the default encoding for text file is different. It does not mean that the user cannot use another encoding, besides the default one, only that it has to be configured. For example, the German umlaut ä on OS/2 is encoded as 132, with Latin1 it is encoded as 228, while in Cyrillic encoding cp1251 this letter does not exist at all. Therefore you should *consider encoding with care* .

The following table shows the default encodings for some operating systems.

Operating system	**Default Encodings**	
	Western Latin	Cyrillic
Modern Unices (*BSD, Mac OS X, GNU/Linux)	`utf-8`	`utf-8`
Mac (before OS X)	`applemac`	`maccyr`
Unix (Old)	`latin1`	`koi8-ru`
Windows	`ansinew` , `cp1252`	`cp1251`
DOS, OS/2	`cp850`	`cp866nav`

UTF-8 and Latin1 are not compatible. It means that if you try to open a Latin1-encoded file using a UTF-8 decoding, it will display odd symbols only if you used accents in it, since both encoding are ASCII superset they encode the *classic* letters the same way. There aren't many advantages in using Latin1 over UTF-8, which is technically superior. UTF-8 is also becoming the most widely used encoding (on the Web, in modern Unices, etc.).

⚠ **Warning**

We really urge you to use UTF-8 encoding. It is technically superior to most (all?) encodings, it supports the full Unicode specification (all symbols that ever existed), and is backward compatible with ASCII. Latin1 is not universal, and having multiple encoding around has always been a source of problems.[3]

11.1.2. Dealing with LaTeX

TeX uses ASCII by default. But 128 characters is not enough to support non-english languages. TeX has its own way to do that with commands for every diacritical marking (see Escaped codes[4]). But if we want accents and other special characters to appear directly in the source file, we have to tell TeX that we want to use a different encoding.

4 Chapter 11.2 on page 126

There are several encodings available to LaTeX:

- ASCII: the default. Only bare english characters are supported in the source file.
- ISO-8859-1 (a.k.a. Latin 1): 8-bits encoding. It supports most characters for latin languages, but that's it.
- UTF-8: a Unicode multi-byte encoding. Supports the complete Unicode specification.
- Others...

In the following we will assume you want to use UTF-8.

There are some *important steps* to specify encoding.

- Make sure your text editor decodes the file in UTF-8.
- Make sure it saves your file in UTF-8. Most text editors do not make the distinction, but some do, such as Notepad++.
- If you are working in a terminal, make sure it is set to support UTF-8 input and output. Some old Unix terminals may not support UTF-8. PuTTY[5] is not set to use UTF-8 by default, you have to configure it.
- Tell LaTeX that the source file is UTF-8 encoded.

```
\usepackage[utf8]{inputenc}
```

`inputenc` [6] package tells LaTeX what the text encoding format of your `.tex` files is.

> ⚠ **Warning**
>
> If you check the character encoding (*e.g.* using the Unix `file` command), be sure that your file contains at least one special character, otherwise it will be recognized as ASCII (which is logical since UTF-8 is as superset of ASCII).

The inputenc package allows as well the user to change the encoding *within the document* by means of the command `\inputencoding{'encoding name'}`.

```
\usepackage[utf8]{inputenc}
% ...
% In this area
% The UTF-8 encoding is specified.
% ...
\inputencoding{latin1}
% ...
% Here the text encoding is specified as ISO Latin-1.
% ...
\inputencoding{utf8}
% Back to the UTF-8 encoding.
% ...
```

5 http://en.wikipedia.org/wiki/PuTTY

6 For a detailed information on the package, see complete specifications written by the package's authors ^{https://www.tug.org/texmf-dist/doc/latex/base/inputenc.pdf} .

11.1.3. Extending the support

The LaTeX support of UTF-8 is fairly specific: it includes only a limited range of unicode input characters. It only defines those symbols that are known to be available with the current *font encoding* . You might encounter a situation where using UTF-8 might result in error:

```
! Package inputenc Error: Unicode char \u8:ũ not set up for use with LaTeX.
```

This is due to the utf8 definition not necessarily having a mapping of all the character glyphs you are able to enter on your keyboard. Such characters are for example

```
ŷ Ŷ ũ Ũ ẽ Ẽ ĩ Ĩ
```

In such case, you may try need to use the `utf8x` option to define more character combinations. `utf8x` is not officially supported, but can be viable in some cases. However it might break up compatibility with some packages like `csquotes`.

Another possiblity is to stick with `utf8` and to define the characters yourself. This is easy:

```
\DeclareUnicodeCharacter{'codepoint'}{'TeX sequence'}
```

where `codepoint` is the unicode codepoint of the desired character. `TeX sequence` is what to print when the character matching the codepoint is met. You may find codepoints on this site[7]. Codepoints are easy to find on the web. Example:

```
\DeclareUnicodeCharacter{0177}{\^y}
```

Now inputting 'ŷ' will effectively print 'ŷ'.

With XeTeX and LuaTeX the inputenc package is no longer needed. Both engines support UTF-8 directly and allow the use of TTF and OpenType fonts to support Unicode characters. See the Fonts[8] section for more information.

11.2. Escaped codes

In addition to direct UTF-8 input, LaTeX supports the composition of special characters. This is convenient if your keyboard lacks some desired accents and other diacritics.

The following accents may be placed on letters. Although 'o' letter is used in most of the examples, the accents may be placed on any letter. Accents may even be placed above a "missing" letter; for example, `\~{}` produces a tilde over a blank space.

The following commands may be used only in paragraph (default) or LR (left-right) mode.

7 `http://www.unicode.org/charts/#symbols`
8 Chapter 9.10 on page 107

LaTeX command	**Sample**	**Description**
`\`{o}`	ò	grave accent
`\'{o}`	ó	acute accent
`\^{o}`	ô	circumflex
`\"{o}`	ö	umlaut, trema or dieresis
`\H{o}`	ő	long Hungarian umlaut (double acute)
`\~{o}`	õ	tilde
`\c{c}`	ç	cedilla
`\k{a}`	ą	ogonek
`\l{}`	ł	barred l (l with stroke)
`\={o}`	ō	macron accent (a bar over the letter)
`\b{o}`	o̲	bar under the letter
`\.{o}`	ȯ	dot over the letter
`\d{u}`	ụ	dot under the letter
`\r{a}`	å	ring over the letter (for å there is also the special command `\aa`)
`\u{o}`	ŏ	breve over the letter
`\v{s}`	š	caron/háček ("v") over the letter
`\t{oo}`	o͡o	"tie" (inverted u) over the two letters
`\o`	ø	slashed o (o with stroke)

To place a diacritic on top of an i or a j, its dot has to be removed. The dotless version of these letters is accomplished by typing `\i` and `\j`. For example:

- `\^\i` should be used for i circumflex 'î';
- `\"\i` should be used for i umlaut 'ï'.

If a document is to be written completely in a language that requires particular diacritics several times, then using the right configuration allows those characters to be written directly in the document. For example, to achieve easier coding of umlauts, the babel package can be configured as `\usepackage[german]{babel}`. This provides the short hand `"o` for `\"o`. This is very useful if one needs to use some text accents in a label, since no backslash will be accepted otherwise.

More information regarding language configuration can be found in the Internationalization[9] section.

11.3. *Less than* < and *greater than* >

The two symbols '<' and '>' are actually ASCII characters, but you may have noticed that they will print '¡' and '¿' respectively. This is a font encoding issue. If you want them to print their real symbol, you will have to use another font encoding such as T1, loaded with the `fontenc` package. See Fonts[10] for more details on font encoding.

9 Chapter 12 on page 133
10 Chapter 9 on page 95

Alternatively, they can be printed with dedicated commands:

```
\textless
\textgreater
```

11.4. Euro € currency symbol

When writing about money these days, you need the euro sign[11]. The `textcomp` package features a `\texteuro` command which gives you the euro symbol as supplied by your current text font. Depending on your chosen font this may be quite far from the official symbol.

An official version of the euro symbol is provided by `eurosym`. Load it in the preamble (optionally with the `official` option):

```
\usepackage[official]{eurosym}
```

then you can insert it with the `\euro{}` command. Finally, if you want a euro symbol that matches with the current font style (e.g., bold, italics, etc.) you can use a different option:

```
\usepackage[gen]{eurosym}
```

again you can insert the euro symbol with `\euro{}`.

Alternatively you can use the `marvosym` package which also provides the official euro symbol.

```
\usepackage{marvosym}
% ...

\EUR{}
```

Now that you have succeeded in printing a euro sign, you may want the '€' on your keyboard to actually print the euro sign as above. There is a simple method to do that. You must make sure you are using UTF-8 encoding along with a working `\euro{}` or `\EUR{}`command.

```
\DeclareUnicodeCharacter{20AC}{\euro{}}
% or
\DeclareUnicodeCharacter{20AC}{\EUR{}}
```

Complete example:

```
\usepackage[utf8]{inputenc}
\usepackage{marvosym}
\DeclareUnicodeCharacter{20AC}{\EUR{}}
```

11.5. Degree symbol for temperature and math

A common mistake is to use the `\circ` command. It will not print the correct character (though `$^\circ$` will). Use the `textcomp` package instead, which provides a `\textdegree` command.

11 `http://en.wikipedia.org/wiki/euro%20sign`

```
\usepackage{textcomp}
%...

A $45$\textdegree angle.
```

For temperature, you can use the same command or opt for the `gensymb` package and write

```
\usepackage{gensymb}
%...

17\degree~C
or
17\celsius
```

Some keyboard layouts feature the degree symbol, you can use it directly if you are using UTF-8 and `textcomp`. For better results (font quality) we recommend the use of an appropriate font, like `lmodern`:

```
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{textcomp}

% ...

17°~C
```

11.6. Other symbols

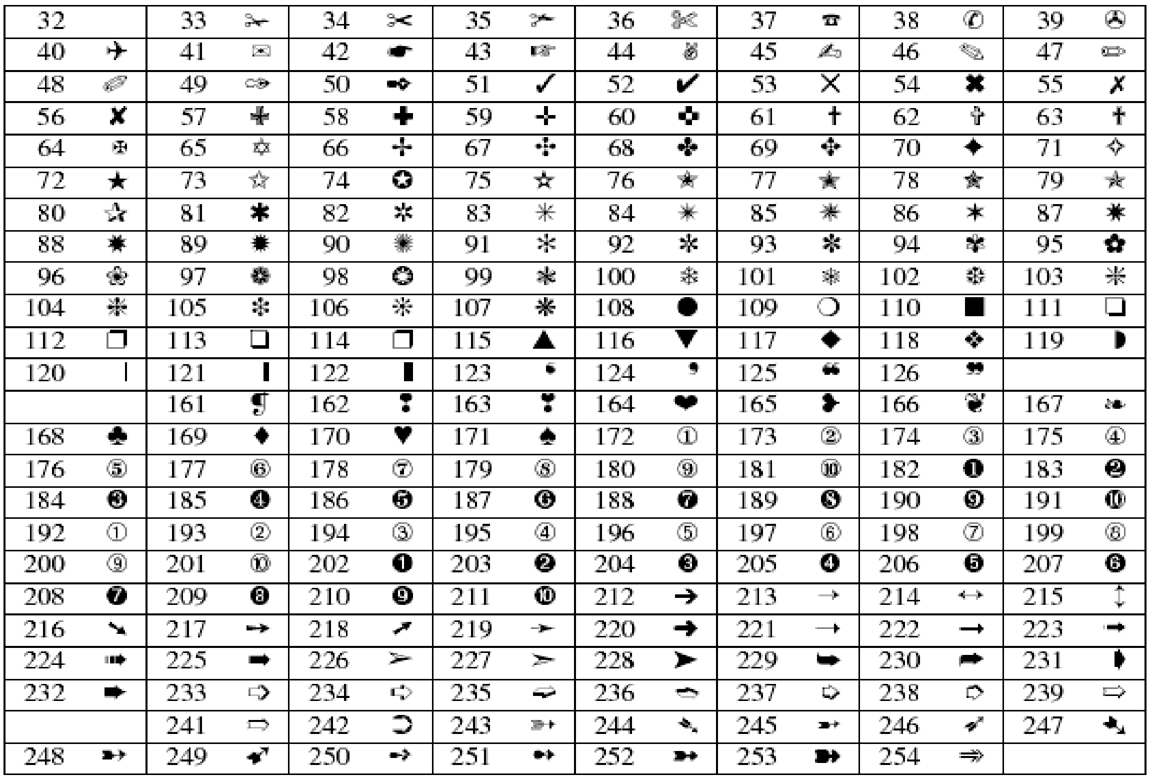
LaTeX has many symbols at its disposal. The majority of them are within the mathematical domain, and later chapters will cover how to get access to them. For the more common text symbols, use the following commands:

Command	**Sample**	**Character**
`\%`	%	%
`\$`	$	$
`\<nowiki>{</nowiki>`	{	{
`\_`	_	_
`\P`	¶	¶
`\ddag`	n/a	‡
`\textbar`	n/a	|
`\textgreater`	>	>
`\textendash`	n/a	–
`\texttrademark`	n/a	™
`\textexclamdown`	n/a	¡
`\textsuperscript<nowiki>{a}</nowiki>`	X^{a}	$^{\text{a}}$
`\pounds`	n/a	£
`\#`	#	#
`\&`	&	&
`\<nowiki>}</nowiki>`	}	}
`\S`	§	§

Command	Sample	Character
`\dag`	n/a	†
`\textbackslash`	n/a	\
`\textless`	<	<
`\textemdash`	n/a	—
`\textregistered`	n/a	®
`\textquestiondown`	n/a	¿
`\textcircled<nowiki>{a}</nowiki>`	n/a	ⓐ
\copyright	n/a	©

Not mentioned in above table, tilde (˜) is used in LaTeX code to produce non-breakable space[12]. To get printed tilde sign, either write `\~{}` or `\textasciitilde{}`. And a visible space ␣ can be created with `\textvisiblespace`.

For some more interesting symbols, the Postscript ZipfDingbats font is available thanks to the `pifont` package. Add the declaration to your preamble: `\usepackage{pifont}`. Next, the command `\ding{number}`, will print the specified symbol. Here is a table of the available symbols:



32		33	✁	34	✂	35	✃	36	✄	37	☎	38	✆	39	✇
40	✈	41	✉	42	☛	43	☞	44	✌	45	✍	46	✎	47	✏
48	✐	49	✑	50	✒	51	✓	52	✔	53	✕	54	✖	55	✗
56	✘	57	✙	58	✚	59	✛	60	✜	61	✝	62	✞	63	✟
64	✠	65	✡	66	✢	67	✣	68	✤	69	✥	70	✦	71	✧
72	★	73	✩	74	✪	75	✫	76	✬	77	✭	78	✮	79	✯
80	✰	81	✱	82	✲	83	✳	84	✴	85	✵	86	✶	87	✷
88	✸	89	✹	90	✺	91	✻	92	✼	93	✽	94	✾	95	✿
96	❀	97	❁	98	❂	99	❃	100	❄	101	❅	102	❆	103	❇
104	❈	105	❉	106	❊	107	❋	108	●	109	❍	110	■	111	❏
112	❐	113	❑	114	❒	115	▲	116	▼	117	◆	118	❖	119	◗
120	❘	121	❙	122	❚	123	❛	124	❜	125	❝	126	❞		
		161	❡	162	❢	163	❣	164	❤	165	❥	166	❦	167	❧
168	♣	169	♦	170	♥	171	♠	172	①	173	②	174	③	175	④
176	⑤	177	⑥	178	⑦	179	⑧	180	⑨	181	⑩	182	❶	183	❷
184	❸	185	❹	186	❺	187	❻	188	❼	189	❽	190	❾	191	❿
192	➀	193	➁	194	➂	195	➃	196	➄	197	➅	198	➆	199	➇
200	➈	201	➉	202	➊	203	➋	204	➌	205	➍	206	➎	207	➏
208	➐	209	➑	210	➒	211	➓	212	➔	213	→	214	↔	215	↕
216	➘	217	➙	218	➚	219	➛	220	➜	221	➝	222	➞	223	➟
224	➠	225	➡	226	➢	227	➣	228	➤	229	➥	230	➦	231	➧
232	➨	233	➩	234	➪	235	➫	236	➬	237	➭	238	➮	239	➯
		241	➱	242	➲	243	➳	244	➴	245	➵	246	➶	247	➷
248	➸	249	➹	250	➺	251	➻	252	➼	253	➽	254	➾		

Figure 36 ZapfDingbats symbols

.

12 Chapter 6.1.3 on page 66

11.7. In special environments

11.7.1. Math mode

Several of the above and some similar accents can also be produced in math mode. The following commands may be used only in math mode.

LaTeX command	Sample	Description	Text-mode equivalence
`\hat{o}`	$\hat{o}$	circumflex	`\^`
`\widehat{oo}`	$\widehat{oo}$	wide version of `\hat` over several letters	
`\check{o}`	$\check{o}$	vee or check	`\v`
`\tilde{o}`	$\tilde{o}$	tilde	`\~`
`\widetilde{oo}`	$\widetilde{oo}$	wide version of `\tilde` over several letters	
`\acute{o}`	$\acute{o}$	acute accent	`\'`
`\grave{o}`	$\grave{o}$	grave accent	``\` ``
`\dot{o}`	$\dot{o}$	dot over the letter	`\.`
`\ddot{o}`	$\ddot{o}$	two dots over the letter (umlaut in text-mode)	`\"`
`\breve{o}`	$\breve{o}$	breve	`\u`
`\bar{o}`	$\bar{o}$	macron	`\=`
`\vec{o}`	$\vec{o}$	vector (arrow) over the letter	

When applying accents to letters `i` and `j`, you can use \imath and \jmath to keep the dots from interfering with the accents:

LaTeX command	Sample	Description	Sample with upper dot
`\hat{\imath}`	$\hat{\imath}$	circumflex on letter `i` without upper dot	$\hat{i}$
`\vec{\jmath}`	$\vec{}$	vector (arrow) on letter `j` without upper dot	$\vec{j}$

11.7.2. Tabbing environment

Some of the accent marks used in running text have other uses in the tabbing environment. In that case they can be created with the following command:

- `\a'` for an acute accent
- ``\a` `` for a grave accent
- `\a=` for a macron accent

11.8. Unicode keyboard input

w:Unicode input[13] Some operating systems provide a keyboard combination to input any Unicode code point, the so-called *unicode compose key* .

Many X applications (*BSD and GNU/Linux) support the `Ctrl+Shift+u` combination. A 'u̲' symbol should appear. Type the code point and press `enter` or `space` to actually print the character. Example:

```
<Ctrl+Shift+u> 20AC <space>
```

will print the euro character.

Desktop environments like GNOME and KDE may feature a customizable compose key for more memorizable sequences.

Xorg features advanced keyboard layouts with variants that let you enter a lot of characters easily with combination using the aprioriate modifier, like `Alt Gr` . It highly depends on the selected layout+variant, so we suggest you to play a bit with your keyboard, preceeding every key and dead key with the `Alt Gr` modifier.

11.9. External links

- A few other LaTeX accents and symbols[14]
- NASA GISS: Accents[15]
- The Comprehensive LATEX Symbol List[16]
- PDF document with a lengthy list of symbols provided by various packages[17]

11.10. Notes and References

13 http://en.wikipedia.org/wiki/Unicode%20input
14 http://spectroscopy.mps.ohio-state.edu/symposium_53/latexinstruct.html
15 http://www.giss.nasa.gov/tools/latex/ltx-401.html
16 http://www.tex.ac.uk/tex-archive/info/symbols/comprehensive/symbols-a4.pdf
17 http://www.rpi.edu/dept/arc/training/latex/LaTeX_symbols.pdf

12. Internationalization

LaTeX has to be configured and used appropriately when it is used to write documents in languages other than English. This has to address three main areas:

1. LaTeX needs to know how to hyphenate the language(s) to be used.
2. The user needs to use language-specific typographic rules. In French for example, there is a mandatory space before each colon character (:).
3. The input of special characters, especially for languages using an input system (Arab, Chinese, Japanese, Korean).

It is convenient to be able to insert language-specific special characters directly from the keyboard instead of using cumbersome coding (for example, by typing `ä` instead of `\"{a}`). This can be done by configuring input encoding properly. We will not tackle this issue here: see the Special Characters[1] chapter.

Some languages require special fonts with the proper font encoding set. See Font encoding[2].

Some of the methods described in this chapter may be useful when dealing with non-English author names in bibliographies.

Here is a collection of suggestions about writing a LaTeX document in a language other than English. If you have experience in a language not listed below, please add some notes about it.

12.1. Prerequisites

Most non-english language will need to input special characters very often. For a convenient writing you will need to set the input encoding and the font encoding properly.

The following configuration is optimal for many languages (most latin languages). Make sure your document is saved using the UTF-8 encoding.

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

⚠ **Warning**

In the following document we will assume you are using this configuration unless otherwise specified.

1 Chapter 11 on page 123
2 Chapter 9.5 on page 99

For more details check Font encoding[3] and Special Characters[4].

12.2. Babel

The `babel` package by Johannes Braams and Javier Bezos will take care of everything (with XeTeX and LuaTeX you should consider `polyglossia`). You can load it in your preamble, providing as an argument name of the language you want to use (usually its English name, but not always):

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

You should place it soon after the `\documentclass` command, so that all the other packages you load afterwards will know the language you are using. Babel will automatically activate the appropriate 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 will disable hyphenation, which has quite a negative effect on the appearance of the typeset document. Babel also specifies new commands for some languages, which simplify the input of special characters. See the sections about languages below for more information.

If you call babel with multiple languages:

```
\usepackage[languageA,languageB]{babel}
```

then the last language in the option list will be active (i.e. languageB), and you can use the command

```
\selectlanguage{languageA}
```

to change the active language. You can also add short pieces of text in another language using the command

```
\foreignlanguage{languageB}{Text in another language}
```

Babel also offers various environments for entering larger pieces of text in another language:

```
\begin{otherlanguage}{languageB}
Text in language B. This environment switches all language-related definitions,
 like the language
specific names for figures, tables etc. to the other language.
\end{otherlanguage}
```

The starred version of this environment typesets the main text according to the rules of the other language, but keeps the language specific string for ancillary things like figures, in the main language of the document. The environment `hyphenrules` switches only the hyphenation patterns used; it can also be used to disallow hyphenation by using the language name 'nohyphenation' (but note `selectlanguage*` is preferred).

The babel manual[5] provides much more information on these and many other options.

3 Chapter 9.5 on page 99
4 Chapter 11 on page 123
5 `http://ftp.snt.utwente.nl/pub/software/tex/macros/latex/required/babel/base/babel.pdf`

12.3. Multilingual versions

It is possible in LaTeX to typeset the content of one document in several languages and to choose upon compilation which language to output. This might be convenient to keep a consistent sectioning and formatting across the different languages. It is also useful if you make use of multiple proper nouns and other untranslated content. Using the commands above in multilingual documents can be cumbersome, and therefore `babel` provides a way to define shorter names. With

```
\babeltags{de = german}
```

You can write:

```
text \textde{German text} text
text
\begin{de}
German text
\end{de}
text
```

12.3.1. Alternative choice using iflang

The current language can also be tested by using the `iflang` package by Heiko Oberdiek (the built-in feature from the babel package is not reliable). Here comes a simple example:

```
\IfLanguageName{ngerman}{Hallo}{Hello}
```

This allows to easily distinguish between two languages without the need of defining own commands. The babel language is changed by setting

```
\selectlanguage{english}
```

12.4. Specific languages

12.4.1. Arabic script

For languages which use the Arabic script, including Arabic, Persian, Urdu, Pashto, Kurdish, Uyghur, etc., add the following code to your preamble:

```
\usepackage{arabtex}
```

You can input text in either romanized characters or native Arabic script encodings. Use any of the following commands and environments to enter in text:

```
\< ... >
\RL{ ... }
\begin{arabtext} ... \end{arabtext}.
```

See the ArabTeX[6] Wikipedia article for further details.

You may also use the `Arabi` package within Babel to typeset Arabic and Persian

```
\usepackage{cmap}
\usepackage[LAE,LFE]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[farsi,arabic]{babel}
```

You may also copy and paste from PDF files produced with Arabi thanks to the support of the `cmap` package. You may use Arabi with LyX, or with tex4ht to produce HTML.

See Arabi page on CTAN[7]

12.4.2. Armenian

The Armenian script uses its own characters, which will require you to install a text editor that supports Unicode[8] and will allow you to enter UTF-8 text, such as Texmaker[9] or WinEdt[10]. These text editors should then be configured to compile using XeLaTeX.

Once the text editor is set up to compile with XeLaTeX, the `fontspec` package can be used to write in Armenian:

```
\usepackage{fontspec}
\setmainfont{DejaVu Serif}
```

or

```
\usepackage{fontspec}
\setmainfont{Sylfaen}
```

The Sylfaen font lacks italic and bold, but DejaVu Serif supports them.

See Armenian Wikibooks[11] for further details, especially on how to configure the Unicode supporting text editors to compile with XeLaTeX.

12.4.3. Cyrillic script

Version 3.7h of `babel` includes support for the `T2*` encodings and for typesetting Bulgarian, Russian and Ukrainian texts using Cyrillic letters[12]. Support for Cyrillic is based on standard LaTeX mechanisms plus the `fontenc` and `inputenc` packages. AMS-LaTeX packages should be loaded before `fontenc` and `babel`. If you are going to use Cyrillics in mathmode, you also need to load `mathtext` package before `fontenc`:

6 `http://en.wikipedia.org/wiki/ArabTeX`
7 `http://www.ctan.org/tex-archive/language/arabic/arabi/`
8 `http://en.wikipedia.org/wiki/Unicode`
9 `http://www.xm1math.net/texmaker/`
10 `http://www.winedt.com/`
11 `http://en.wikibooks.org/wiki/%3Ahy%3A%D4%BC%D5%A1%D5%8F%D5%A5%D4%BD%2F%D4%B2%D5%A1%D6%80%D6%87%20%D5%A1%D5%B7%D5%AD%D5%A1%D6%80%D5%B0`
12 The Not So Short Introduction to LaTeX ˆ{`http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf`} , 2.5.6 Support for Cyrillic, Maksym Polyakov

```
\usepackage{amsmath,amsthm,amssymb}
\usepackage{mathtext}

\usepackage[T1,T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[english,bulgarian,russian,ukrainian]{babel}
```

Generally, `babel` will automatically choose the default font encoding, for the above three languages this is `T2A`. However, documents are not restricted to a single font encoding. For multilingual documents using Cyrillic and Latin-based languages it makes sense to include Latin font encoding explicitly. Babel will take care of switching to the appropriate font encoding when a different language is selected within the document.

On modern operating systems it is beneficial to use Unicode (`utf8` or `utf8x`) instead of KOI8-RU (`koi8-ru`) as an input encoding for Cyrillic text.

In addition to enabling hyphenations, translating automatically generated text strings, and activating some language specific typographic rules (like `\frenchspacing`), `babel` provides some commands allowing typesetting according to the standards of Bulgarian, Russian, or Ukrainian languages.

For all three languages, language specific punctuation is provided: the Cyrillic dash for the text (it is little narrower than Latin dash and surrounded by tiny spaces), a dash for direct speech, quotes, and commands to facilitate hyphenation:

Key combination	**Action**
`"|`	No ligature at this position.
`"-`	Explicit hyphen sign, allowing hyphenation in the rest of the word.
`"---`	Cyrillic emdash in plain text.
`"--~`	Cyrillic emdash in compound names (surnames).
`"--*`	Cyrillic emdash for denoting direct speech.
`""`	Similar to `"-`, but it produces no hyphen sign (used for compound words with hyphen, e.g. `x-""y` or some other signs as “disable/enable”).
`"~`	Compound word mark without a breakpoint.
`"=`	Compound word mark with a breakpoint, allowing hyphenation in the composing words.
`",`	Thinspace for initials with a breakpoint in a following surname.
`"‘`	German opening double quote („).
`"’`	German closing double quote (“).
`"<`	French opening double quote (<<).
`">`	French closing double quote (>>).

The Russian and Ukrainian options of `babel` define the commands

```
\Asbuk
\asbuk
```

which act like `\Alph` and `\alph` (commands for turning counters into letters, *e.g.* `a, b, c...`), but produce capital and small letters of Russian or Ukrainian alphabets (whichever is the active language of the document).

The Bulgarian option of `babel` provides the commands

```
\enumBul
\enumLat
\enumEng
```

which make `\Alph` and `\alph` produce letters of either Bulgarian or Latin (English) alphabets. The default behaviour of `\Alph` and `\alph` for the Bulgarian language option is to produce letters from the Bulgarian alphabet.

See the Bulgarian translation of ”The Not So Short Introduction to LaTeX” [13] for a method to type Cyrillic letters directly from the keyboard using a different distribution.

12.4.4. Chinese

One possible Chinese support is made available thanks to the `CJK` package collection. If you are using a package manager or a portage tree, the CJK collection is usually in a separate package because of its size (mainly due to fonts).

Make sure your document is saved using the UTF-8 character encoding. See Special Characters[14] for more details. Put the parts where you want to write chinese characters in a `CJK` environment.

```
\documentclass{article}
\usepackage{CJK}

\begin{document}

\begin{CJK}{UTF8}{gbsn}
⊠⊠
You can mix latin letters and chinese.
\end{CJK}

\end{document}
```

The last argument specifies the font. It must fit the desired language, since fonts are different for Chinese, Japanese and Korean. Possible choices for Chinese include:

- gbsn (简体宋体, simplified Chinese)
- gkai (简体楷体, simplified Chinese)
- bsmi (繁体细上海宋体, traditional Chinese)
- bkai (繁体标楷体, traditional Chinese)

12.4.5. Czech

Czech is fine using

13 The Not So Short Introduction to LaTeX ˆ{`http://www.ctan.org/tex-archive/info/lshort/bulgarian/lshort-bg.pdf`} , Bulgarian translation

14 Chapter 11 on page 123

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

UTF-8 allows you to have „czech quotation marks“ directly in your text. Otherwise, there are macros *\clqq* and *\crqq* to produce left and right quote. You can place quotated text inside `\uv`.

12.4.6. Finnish

Finnish language hyphenation is enabled with:

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

This will also automatically change document language (section names, etc.) to Finnish.

12.4.7. French

You can load French language support with the following command:

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

There are multiple options for typesetting French documents, depending on the flavor of French: `french`, `frenchb`, and `francais` for Parisian French, and `acadian` and `canadien` for new-world French. If you do not know or do not really care, we would recommend using `frenchb`.

All enable French hyphenation, if you have configured your LaTeX system accordingly. All of these also change all automatic text into French: `\chapter` prints *Chapitre* , `\today` prints the current date in French and so on. A set of new commands also becomes available, which allows you to write French input files more easily. Check out the following table for inspiration:

input code	**rendered output**
`\og guillemets \fg{}`	« guillemets »
`M\up{me}, D\up{r}`	M^{me}, D^{r}
`1\ier{}, 1\iere{}, 1\ieres{}`	1er, 1re, 1res
`2\ieme{} 4\iemes{}`	2^{e} 4es
`\No 1, \no 2`	N° 1, n° 2
`20~\degres C, 45\degres`	20 °C, 45°
`M. \bsc{Durand}`	M. Durand
`\nombre{1234,56789}`	1 234,567 89

You may want to typeset guillemets and other French characters directly if your keyboard have them. Running Xorg (*BSD and GNU/Linux), you may want to use the `oss` variant which features some nice shortcuts, like

Key combination	**Character**
`Alt Gr + w`	«
`Alt Gr + x`	»

Key combination	**Character**
`Alt Gr + Shift + é`	É
`Alt Gr + Shift + è`	È
`Alt Gr + Shift + ç`	Ç

You will need the T1 font encoding for guillemets to print properly.

For the degree character you will get an error like

```
! Package inputenc Error: Unicode char \u8:° not set up for use with LaTeX.
```

The `textcomp` package will fix it for you.

The great advantage of Babel for French is that it will handle some elements of French typography for you, especially non-breaking spaces before all two-parts punctuation marks. So now you can write:

```
Il répondit: «Ce pain coûte-t-il 2~€?»
```

The non-breaking space before the euro symbol is still necessary because currency symbols and other units or not supported in general (that's not specific to French).

You can use the `numprint` package along Babel. It will let you print numbers the French way.

```
\usepackage[frenchb]{babel}
\usepackage[autolanguage]{numprint} % Must be loaded *after* babel.

% ...

\nombre{123456.123456 e-17}
```

$123\,456{,}123\,456 \cdot 10^{-17}$

You will also notice that the layout of lists changes when switching to the French language. This is customizable using the `\frenchbsetup` command. For more information on what the `frenchb` option of `babel` does and how you can customize its behavior, run LaTeX on file `frenchb.dtx` and read the produced file `frenchb.pdf` or `frenchb.dvi` . You can get the PDF version on CTAN[15].

12.4.8. German

You can load German language support using *either one* of the two following commands.

For traditional ("old") German orthography use

15 `http://mirrors.ctan.org/macros/latex/contrib/babel-contrib/frenchb/frenchb.pdf`

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

or for reform (”new”) German orthography use

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

This enables German hyphenation, if you have configured your LaTeX system accordingly. It also changes all automatic text into German, e.g. “Chapter” becomes “Kapitel”. A set of new commands also becomes available, which allows you to write German input files more quickly even when you don’t use the inputenc package. Check out the table below for inspiration. With inputenc, all this becomes moot, but your text also is locked in a particular encoding world.

German Special Characters.	
"A "O "U	Ä Ö Ü
"a "o "u "s	ä ö ü ß
"` or \glqq	„
"' or \grqq	“
\glq \grq	
"< or \flqq	«
"> or \frqq	»
\flq \frq	‹ ›
\dq	”

In German books you sometimes find French quotation marks («guillemets»). German typesetters, however, use them differently. A quote in a German book would look like »this«. In the German speaking part of Switzerland, typesetters use «guillemets» the same way the French do. A major problem arises from the use of commands like `\flq`: If you use the OT1 font encoding (which is the default) the guillemets will look like the math symbol ”$\ll$”, which turns a typesetter’s stomach. T1 encoded fonts, on the other hand, do contain the required symbols. So if you are using this type of quote, make sure you use the T1 encoding.

Decimal numbers usually have to be written like `0{,}5` (not just 0,5). Packages like `ziffer` enable input like `0,5` .

12.4.9. Greek

This is the preamble you need to write in the Greek language. Note the particular input encoding.

```
\usepackage[english,greek]{babel}
\usepackage[iso-8859-7]{inputenc}
```

This preamble enables hyphenation and changes all automatic text to Greek. A set of new commands also becomes available, which allows you to write Greek input files more easily. In order to temporarily switch to English and vice versa, one can use the commands `\text-latin{english text}` and `\textgreek{greek text}` that both take one argument which is then typeset using the requested font encoding. Otherwise you can use the command `\selectlanguage{...}` described in a previous section. Use `\euro` for the Euro symbol.

12.4.10. Hungarian

Use the following lines:

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

More information in hungarian[16].

12.4.11. Icelandic and Faroese

The following lines can be added to write Icelandic text:

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

This changes text like *Part* into *Hluti* . It makes additional commands available:

Icelandic special commands	
`"`` or `\glqq`	„
`\grqq`	“
`\TH`	Þ
`\th`	þ
`\DH`	Đ
`\dh`	ð

To make special characters such as Þ and Æ become available just add:

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

The default LATEX font encoding is OT1, but it contains only the 128 characters. The T1 encoding contains letters and punctuation characters for most of the European languages using Latin script.

12.4.12. Italian

Italian is well supported by LaTeX. Just add

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

at the beginning of your document and the output of all the commands will be translated properly.

16 `http://www.math.bme.hu/latex/`

12.4.13. Japanese

There is a variant of TeX intended for Japanese named pTeX[17], which supports vertical typesetting.

Another possible way to write in japanese is to use Lualatex and the luatex-ja package. An example from the Luatexja documentation :

```
\documentclass{ltjsarticle}
\usepackage{luatexja} % ltjclasses, ltjsclasses ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
\begin{document}
\section{⊠⊠⊠⊠⊠Lua\TeX-ja}
⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
\subsection{⊠⊠⊠⊠⊠}
⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
\end{document}
```

You can also use capabilities provided by the Fontspec package and those provided by Luatexja-fontspec to declare the font you want to use in your paper. Let us take an example :

```
% *********************************
% Basic setup
\documentclass[10pt,a4paper]{article}
\usepackage{luatextra}%this package calls fontspec, luatexbase, lualibs,
 metalogo, luacode and fixltx2e
\setmainfont[Ligatures=Rare,Numbers=OldStyle]{Arno Pro} %setup of western font
\usepackage{luatexja}
\usepackage{luatexja-fontspec}%needed to call \setmainjfont bellow
\setmainjfont[BoldFont=KozGoPr6N-Bold]{KozGoPr6N-Regular} %setup of japanese
 font
%*********************************
\begin{document}
It is a test to show japanese and english mix. ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
\end{document}
```

Use UTF-8 as your encoding. In case you don't know how to do this, take a look at Texmaker, a LaTeX editor which use UTF-8 by default.

Another (but old) possible Japanese support is made available thanks to the `CJK` package collection. If you are using a package manager or a portage tree, the CJK collection is usually in a separate package because of its size (mainly due to fonts).

Make sure your document is saved using the UTF-8 character encoding. See Special Characters[18] for more details. Put the parts where you want to write japanese characters in a `CJK` environment.

```
\documentclass{article}
\usepackage{CJK}

\begin{document}

\begin{CJK}{UTF8}{min}
⊠⊠⊠⊠⊠
You can mix latin letters as well as hiragana, katakana and kanji.
\end{CJK}
```

17 `http://ascii.asciimw.jp/pb/ptex/`
18 Chapter 11 on page 123

```
\end{document}
```

The last argument specifies the font. It must fit the desired language, since fonts are different for Chinese, Japanese and Korean. `min` is an example for Japanese.

12.4.14. Korean

The two most widely used encodings for Korean text files are EUC-KR and its upward compatible extension used in Korean MS-Windows, CP949/Windows-949/UHC. In these encodings each US-ASCII character represents its normal ASCII character similar to other ASCII compatible encodings such as ISO-8859-x, EUC-JP, Big5, or Shift_JIS. On the other hand, Hangul syllables, Hanjas (Chinese characters as used in Korea), Hangul Jamos, Hiraganas, Katakanas, Greek and Cyrillic characters and other symbols and letters drawn from KS X 1001 are represented by two consecutive octets. The first has its MSB set. Until the mid-1990's, it took a considerable amount of time and effort to set up a Korean-capable environment under a non-localized (non-Korean) operating system. You can skim through the now much-outdated `http://jshin.net/faq` to get a glimpse of what it was like to use Korean under non-Korean OS in mid-1990's.

TeX and LaTeX were originally written for scripts with no more than 256 characters in their alphabet. To make them work for languages with considerably more characters such as Korean or Chinese, a subfont mechanism was developed. It divides a single CJK font with thousands or tens of thousands of glyphs into a set of subfonts with 256 glyphs each.

For Korean, there are three widely used packages.

- HLATEX by UN Koaunghi
- hLATEXp by CHA Jaechoon
- the CJK package by Werner Lemberg

HLATEX and hLATEXp are specific to Korean and provide Korean localization on top of the font support. They both can process Korean input text files encoded in EUC-KR. HLATEX can even process input files encoded in CP949/Windows-949/UHC and UTF-8 when used along with Λ, Ω.

The CJK package is not specific to Korean. It can process input files in UTF-8 as well as in various CJK encodings including EUC-KR and CP949/Windows-949/UHC, it can be used to typeset documents with multilingual content (especially Chinese, Japanese and Korean). The CJK package has no Korean localization such as the one offered by HLATEX and it does not come with as many special Korean fonts as HLATEX.

The ultimate purpose of using typesetting programs like TeX and LaTeX is to get documents typeset in an *aesthetically* satisfying way. Arguably the most important element in typesetting is a set of welldesigned fonts. The HLATEX distribution includes UHC PostScript fonts of 10 different families and Munhwabu fonts (TrueType) of 5 different families. The CJK package works with a set of fonts used by earlier versions of HLATEX and it can use Bitstream's cyberbit True-Type font.

To use the HLATEX package for typesetting your Korean text, put the following declaration into the preamble of your document:

```
\usepackage{hangul}
```

This command turns the Korean localization on. The headings of chapters, sections, subsections, table of content and table of figures are all translated into Korean and the formatting of the document is changed to follow Korean conventions. The package also provides automatic *particle selection* . In Korean, there are pairs of post-fix particles grammatically equivalent but different in form. Which of any given pair is correct depends on whether the preceding syllable ends with a vowel or a consonant. (It is a bit more complex than this, but this should give you a good picture.) Native Korean speakers have no problem picking the right particle, but it cannot be determined which particle to use for references and other automatic text that will change while you edit the document. It takes a painstaking effort to place appropriate particles manually every time you add/remove references or simply shuffle parts of your document around. HLATEX relieves its users from this boring and error-prone process.

In case you don't need Korean localization features but just want to typeset Korean text, you can put the following line in the preamble, instead.

```
\usepackage{hfont}
```

For more details on typesetting Korean with HLATEX, refer to the HLATEX Guide. Check out the web site of the Korean TeX User Group (KTUG)[19].

In the FAQ section of KTUG it is recommended to use the kotex package

```
\usepackage{kotex}
```

12.4.15. Persian script

For Persian language, there is a dedicated package called XePersian which uses XeLaTeX as the typesetting engine. Just add the following code to your preamble:

```
\usepackage{xepersian}
```

See XePersian page on CTAN[20]

Moreover, Arabic script can be used to type Persian as illustrated in the corresponding section[21].

12.4.16. Polish

If you plan to use Polish in your UTF-8 encoded document, use the following code

```
\usepackage[utf8]{inputenc}
\usepackage{polski}
\usepackage[polish]{babel}
```

19 `http://www.ktug.or.kr/`
20 `http://www.ctan.org/tex-archive/macros/xetex/latex/xepersian/`
21 Chapter 12.4.20 on page 149

The above code merely allows to use Polish letters and translates the automatic text to Polish, so that ”chapter” becomes ”rozdział”. There are a few additional things one must remember about.

Connectives

Polish has many single letter connectives: ”a”, ”o”, ”w”, ”i”, ”u”, ”z”, etc., grammar and typography rules don’t allow for them to end a printed line. To ensure that LaTeX won’t set them as last letter in the line, you have to use non breakable space:

```
Noc była sierpniowa, ciepła i~słodka, Księżyc oświecał srebrnem światłem
 wgłębienie, tak,
że twarze małego rycerza i~Basi były skąpane w blasku.
Poniżej, na podwórzu zamkowem, widać było uśpione kupy żołnierzy, a~także
 i~ciała zabitych
podczas dziennej strzelaniny, bo nie znaleziono dotąd czasu na ich pogrzebanie.
```

Numerals

According to Polish grammar rules, you have to put dots after numerals in chapter, section, subsection, etc. headers.

This is achieved by redefining few LaTeX macros.

For books:

```
\renewcommand\thechapter{\arabic{chapter}.}
\renewcommand\thesection{\arabic{chapter}.\arabic{section}.}
\re
newcommand\thesubsection{\arabic{chapter}.\arabic{section}.\arabic{subsection}.}
\renewcommand\thesubsubsectio
n{\arabic{chapter}.\arabic{section}.\arabic{subsection}.\arabic{subsubsection}.}
```

For articles:

```
\renewcommand\thesection{\arabic{section}.}
\renewcommand\thesubsection{\arabic{section}.\arabic{subsection}.}
\renewcomman
d\thesubsubsection{\arabic{section}.\arabic{subsection}.\arabic{subsubsection}.}
```

Alternatively you can use dedicated document classes:

- the `mwart` class instead of `article`,
- `mwbk` instead of `book`
- and `mwrep` instead of `report`.

Those classes have much more European typography settings but *do not* require the use of Polish babel settings or character encoding.

Simple usage:

```
\documentclass{mwart}
\usepackage[polish]{babel}
\usepackage{polski}
\begin{document}
```

```
Pójdź kińże tę chmurność w głąb flaszy.
\end{document}
```

Full documentation for those classes is available at `http://web.archive.org/web/20040609034031/http://www.ci.pwr.wroc.pl/~pmazur/LaTeX/mwclsdoc.pdf` (Polish).

Indentation

It may be customary (depending on publisher) to indent the first paragraph in sections and chapters:

```
\usepackage{indentfirst}
```

Hyphenation and typography

It's much more frowned upon to set pages with hyphenation between pages than it is customary in American typesetting.

To adjust penalties for hyphenation spanning pages, use this command:

```
\brokenpenalty=1000
```

To adjust penalties for leaving widows and orphans (clubs in TeX nomenclature) use those commands:

```
\clubpenalty=1000
\widowpenalty=1000
```

Commas in math

According to Polish typography rules, fractional parts of numbers should be delimited by a comma, not a dot. To make LaTeX not insert additional space in math mode after a comma (unless there is a space after the comma), use the `icomma` package.

```
\usepackage{icomma}
```

Unfortunately, it is partially incompatible with the `dcolumn` package. One needs to either use dots in columns with numerical data in the source file and make `dcolumn` switch them to commas for display *or* define the column as follows:

```
\begin{tabular}{... D{,}{\mathord\mathcomma}{2} ...}
```

The alternative is to use the `numprint` package, but it is much less convenient.

Further information

Refer the Słownik Ortograficzny[22] (in Polish) for additional information on Polish grammar and typography rules.

22 `http://so.pwn.pl/zasady.php`

Good extract is available at Zasady Typograficzne Składania Tekstu[23] (in Polish).

12.4.17. Portuguese

Add the following code to your preamble:

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

You can substitute the language for brazilian portuguese by choosing `brazilian` or `brazil`.

12.4.18. Slovak

Basic settings are fine when left the same as Czech, but Slovak needs special signs for 'ď', 'ť', 'ľ'. To be able to type them from keyboard use the following settings:

```
\usepackage[slovak]{babel}
\usepackage[IL2]{fontenc}
```

12.4.19. Spanish

Include the appropriate Babel option:

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

The trick is that Spanish has several options and commands to control the layout. The options may be loaded either at the call to Babel, or before, by defining the command `\spanishoptions`. Therefore, the following commands are roughly equivalent:

```
\def\spanishoptions{mexico}
\usepackage[spanish]{babel}
```

```
\usepackage[spanish,mexico]{babel}
```

On average, the former syntax should be preferred, as the latter is a deviation from standard Babel behavior, and thus may break other programs (LyX, latex2rtf) interacting with LaTeX.

Spanish also defines shorthands for the dot and << >> so that they are used as logical markup: the former is used as decimal marker in math mode, and the output is typically either a comma or a dot; the latter is used for quoted text, and the output is typically either «» or “”. This allows different typographical conventions with the same input, as preferences may be quite different from, say, Spain and Mexico.

Two particularly useful options are `es-noquoting,es-nolists`: some packages and classes are known to collide with Spanish in the way they handle active characters, and these options disable the internal workings of Spanish to allow you to overcome these common pitfalls. Moreover, these options may simplify the way LyX customizes some features of the Spanish layout from inside the GUI.

23 `http://dtp.msstudio.com.pl/typo.html`

The options `mexico,mexico-com` provide support for local custom in Mexico: the former using decimal dot, as customary, and the latter allowing decimal comma, as required by the Mexican Official Norm (NOM) of the Department of Economy for labels in foods and goods. More localizations are in the making.

The other commands modify the spanish layout after loading Babel. Two particularly useful commands are `\spanishoperators` and `\spanishdeactivate`.

The macro `\spanishoperators{<list of operators>}{` contains a list of spanish mathematical operators, and may be redefined at will. For instance, the command

```
\def\spanishoperators{sen}
```

only defines `sen`, overriding all other definitions; the command `\let\spanishoperators\relax` disables them all. This command supports accented or spaced operators: the `\acute{<letter>}` command puts an accent, and the `\,` command adds a small space. For instance, the following operators are defined by default.

```
l\acute{i}m l\acute{i}m\,sup l\acute{i}m\,inf m\acute{a}x
\acute{i}nf m\acute{i}n sen tg arc\,sen arc\,cos arc\,tg
cotg cosec senh tgh
```

Finally, the macro `\spanishdeactivate{<list of characters>}` disables some active characters, to keep you out of trouble if they are redefined by other packages. The candidates for deactivation are the set {<>."'}. Please, beware that some option preempt the availability of some active characters. In particular, you should not combine the `es-noquoting` option with `\spanishdeactivate{<>}`, or the `es-noshorthands` with `\spanishdeactivate{<>."}`.

Please check the documentation for Babel or `spanish.dtx` for further details.

12.4.20. Tibetan

One option to use Tibetan script in LaTeX is to add

```
\usepackage{ctib}
```

to your preamble and use a slightly modified Wylie transliteration for input. Refer to the excellent package documentation for details. More information can be found on `http://www.thlib.org/tools/scripts/wiki/latex.html`

12.5. References

13. Rotations

⚠ **Warning**

Many DVI viewers do not support rotating of text and tables. The text will be displayed normally. You must convert your DVI file to a PDF document and view it in a PDF viewer to see the rotation in effect. Take care however that printing from those PDF files may rotate the respective page *again in the same direction* under certain circumstances. This behaviour can be influenced by the settings of your dvi2pdf converter, look at your manual for further information.

13.1. The *rotating* package

The package `rotating` gives you the possibility to rotate any object of an arbitrary angle. Once you have loaded it with the standard command in the preamble:

```
\usepackage{rotating}
```

you can use three new environments:

```
\begin{sideways}
```

it will rotate the whole argument by 90 degrees counterclockwise. Moreover:

```
\begin{turn}{30}
```

it will turn the argument of 30 degrees. You can give any angle as an argument, whether it is positive or negative. It will leave the necessary space to avoid any overlapping of text.

```
\begin{rotate}{30}
```

like `turn`, but it will not add any extra space.

If you want to make a float sideways so that the caption is also rotated, you can use

```
\begin{sidewaysfigure}
```

or

```
\begin{sidewaystable}
```

Note, though, they will be placed on a separate page.

If you would like to rotate a TikZ picture you could use `sideways` together with `minipage`.

```
\begin{figure}[htbp]
  \begin{sideways}
    \begin{minipage}{17.5cm}
      \input{../path/to/picture}
    \end{minipage}
  \end{sideways}
  \centering
  \caption[Caption]{Caption.}
  \label{pic:picture}
\end{figure}
```

You can also use the `\rotatebox` command. Let's rotate a tabular inside a table for example:

```
\begin{table}[p]
        \centering
        \rotatebox{90}{
                \begin{minipage}{\textheight}
                \begin{tabular}{
```

13.1.1. Options

Default is sidewaysfigures/sidewaystables are oriented depending on page number in two sided documents (takes two passes).

The rotating package takes the following options.

counterclockwise/anticlockwise

In single sided documents turn sidewaysfigures/sidewaystables counterclockwise.

clockwise

In single sided documents turn sidewaysfigures/sidewaystables clockwise (default).

figuresright

In two sided documents all sidewaysfigures/sidewaystables are same orientation (left of figure, table now bottom of page). This is the style preferred by the Chicago Manual of Style (broadside).

figuresleft

In two sided documents all sidewaysfigures/sidewaystables are same orientation (left of figure, table now at top of page).

13.2. The *rotfloat* package

When it is desirable to place the rotated table at the exact location where it appears in the source (.tex) file, `rotfloat` package may be used. Then one can use

```
\begin{sidewaystable}[H]
```

just like for normal tables. The `H` option can not be used without this package.

14. Tables

Tables are a common feature in academic writing, often used to summarize research results. Mastering the art of table construction in LaTeX is therefore necessary to produce quality papers and with sufficient practice one can print beautiful tables of any kind.

Keeping in mind that LaTeX is not a spreadsheet, it makes sense to use a dedicated tool to build tables and then to export these tables into the document. Basic tables are not too taxing, but anything more advanced can take a fair bit of construction; in these cases, more advanced packages can be very useful. However, first it is important to know the basics. Once you are comfortable with basic LaTeX tables, you might have a look at more advanced packages or the export options of your favorite spreadsheet[1]. Thanks to the modular nature of LaTeX, the whole process can be automated in a fairly comfortable way.

For a long time, LaTeX tables were quite a chaotic topic, with dozens of packages doing similar things, while not always being compatible with one another. Sometimes you had to make trade-offs. The situation changed recently (2010) with the release of the `tabu` package which combines the power of `longtable`, `tabularx` and much more. The `tabu` environment is far less fragile and restricted than the older alternatives. Nonetheless, before attempting to use this package for the first time it will be beneficial to understand how the classic environment works, since `tabu` works the same way. Note however that the author of `tabu` will not fix bugs to the current version, and that the next version introduces new syntax that will likely break existing documents.[2]

14.1. The *tabular* environment

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

The first line of the environment has the form:

```
\begin{tabular}[pos]{table spec}
```

The `table spec` argument tells LaTeX the alignment to be used in each column and the vertical lines to insert.

The number of columns does not need to be specified as it is inferred by looking at the number of arguments provided. It is also possible to add vertical lines between the columns here. The following symbols are available to describe the table columns (some of them require that the package `array` has been loaded):

1 Chapter 14.16 on page 185
2 `http://tex.stackexchange.com/questions/121841/is-the-tabu-package-obsolete`

`l`	left-justified column
`c`	centered column
`r`	right-justified column
`p{'width'}`	paragraph column with text vertically aligned at the top
`m{'width'}`	paragraph column with text vertically aligned in the middle (requires array package)
`b{'width'}`	paragraph column with text vertically aligned at the bottom (requires array package)
`|`	vertical line
`||`	double vertical line

By default, if the text in a column is too wide for the page, LaTeX won't automatically wrap it. Using `p{'width'}` you can define a special type of column which will wrap-around the text as in a normal paragraph. You can pass the width using any unit supported by LaTeX, such as 'pt' and 'cm', or *command lengths* , such as `\textwidth`. You can find a list in chapter Lengths[3].

The optional parameter `pos` can be used to specify the vertical position of the table relative to the baseline of the surrounding text. In most cases, you will not need this option. It becomes relevant only if your table is not in a paragraph of its own. You can use the following letters:

`b`	bottom
`c`	center (default)
`t`	top

To specify a font format (such as bold, italic, etc.) for an entire column, you can add `>{\format}` before you declare the alignment. For example `\begin{tabular}{ >{\bf-series}l c >{\itshape}r }` will indicate a three column table with the first one aligned to the left and in bold font, the second one aligned in the center and with normal font, and the third aligned to the right and in italic. The "array" package needs to be activated in the preamble for this to work.

In the first line you have pointed out how many columns you want, their alignment and the vertical lines to separate them. Once in the environment, you have to introduce the text you want, separating between cells and introducing new lines. The commands you have to use are the following:

`&`	column separator
`\\`	start new row (additional space may be specified after `\\` using square brackets, such as `\\[6pt]`)
`\hline`	horizontal line
`\newline`	start a new line within a cell (in a paragraph column)
`\cline{i-j}`	partial horizontal line beginning in column i and ending in column j

3 Chapter 23 on page 283

Note, any white space inserted between these commands is purely down to ones' preferences. I personally add spaces between to make it easier to read.

14.1.1. Basic examples

This example shows how to create a simple table in LaTeX. It is a three-by-three table, but without any lines.

```
\begin{tabular}{ l c r }
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

1	2	3
4	5	6
7	8	9

Expanding upon that by including some vertical lines:

```
\begin{tabular}{ l  c  r }
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

1	2	3
4	5	6
7	8	9

To add horizontal lines to the very top and bottom edges of the table:

```
\begin{tabular}{ l  c  r }
  \hline
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
  \hline
\end{tabular}
```

1	2	3
4	5	6
7	8	9

And finally, to add lines between all rows, as well as centering (notice the use of the center environment - of course, the result of this is not obvious from the preview on this web page):

```
\begin{center}
  \begin{tabular}{ l  c  r }
    \hline
    1 & 2 & 3 \\ \hline
    4 & 5 & 6 \\ \hline
    7 & 8 & 9 \\
    \hline
  \end{tabular}
\end{center}
```

1	2	3
4	5	6
7	8	9

```
\begin{center}
  \begin{tabular}{  l  c  r }
    \hline
    1 & 2 & 3 \\ \hline
    4 & 5 & 6 \\ \hline \hline
    7 & 8 & 9 \\
    \hline
  \end{tabular}
\end{center}
```

1	2	3
4	5	6
7	8	9

```
\begin{tabular}
  \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

Figure 37

14.1.2. Text wrapping in tables

LaTeX's algorithms for formatting tables have a few shortcomings. One is that it will not automatically wrap text in cells, even if it overruns the width of the page. For columns that will contain text whose length exceeds the column's width, it is recommended that you use the `p` attribute and specify the desired width of the column (although it may take some trial-and-error to get the result you want). For a more convenient method, have a look at The tabularx package[4], or The tabulary package[5].

Instead of `p`, use the `m` attribute to have the lines aligned toward the middle of the box or the `b` attribute to align along the bottom of the box.

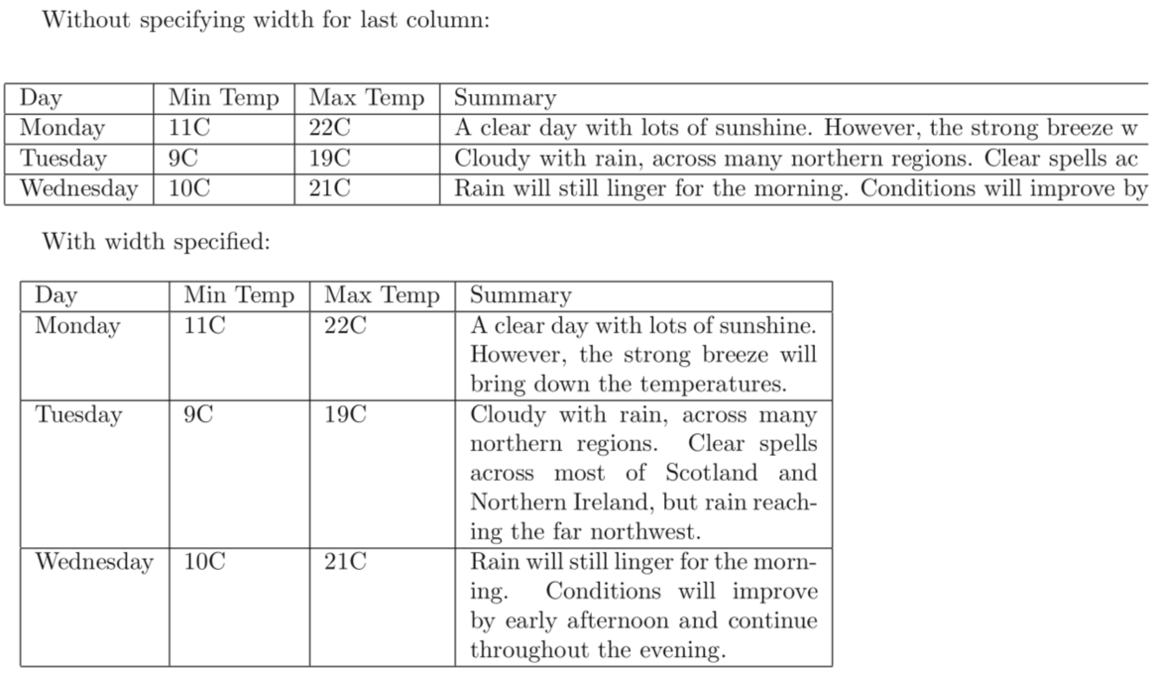
Here is a simple example. The following code creates two tables with the same code; the only difference is that the last column of the second one has a defined width of 5 centimeters, while in the first one we didn't specify any width. Compiling this code:

```
\documentclass{article}
\usepackage[english]{babel}

\begin{document}

Without specifying width for last column:
\begin{center}
    \begin{tabular} l  l  l  l }
    \hline
    Day & Min Temp & Max Temp & Summary \\ \hline
    Monday & 11C & 22C & A clear day with lots of sunshine.
    However, the strong breeze will bring down the temperatures. \\ \hline
    Tuesday & 9C & 19C & Cloudy with rain, across many northern regions. Clear
 spells
    across most of Scotland and Northern Ireland,
    but rain reaching the far northwest. \\ \hline
    Wednesday & 10C & 21C & Rain will still linger for the morning.
    Conditions will improve by early afternoon and continue
    throughout the evening. \\
    \hline
    \end{tabular}
\end{center}

With width specified:
\begin{center}
    \begin{tabular}{  l  l  l  p{5cm} }
```

4 Chapter 14.6 on page 173
5 Chapter 14.6 on page 173

```
    \hline
    Day & Min Temp & Max Temp & Summary \\ \hline
    Monday & 11C & 22C & A clear day with lots of sunshine.
    However, the strong breeze will bring down the temperatures. \\ \hline
    Tuesday & 9C & 19C & Cloudy with rain, across many northern regions. Clear
 spells
    across most of Scotland and Northern Ireland,
    but rain reaching the far northwest. \\ \hline
    Wednesday & 10C & 21C & Rain will still linger for the morning.
    Conditions will improve by early afternoon and continue
    throughout the evening. \\
    \hline
    \end{tabular}
\end{center}

\end{document}
```

You get the following output:



Without specifying width for last column:

Day	Min Temp	Max Temp	Summary
Monday	11C	22C	A clear day with lots of sunshine. However, the strong breeze w
Tuesday	9C	19C	Cloudy with rain, across many northern regions. Clear spells ac
Wednesday	10C	21C	Rain will still linger for the morning. Conditions will improve by

With width specified:

Day	Min Temp	Max Temp	Summary
Monday	11C	22C	A clear day with lots of sunshine. However, the strong breeze will bring down the temperatures.
Tuesday	9C	19C	Cloudy with rain, across many northern regions. Clear spells across most of Scotland and Northern Ireland, but rain reaching the far northwest.
Wednesday	10C	21C	Rain will still linger for the morning. Conditions will improve by early afternoon and continue throughout the evening.

Figure 38

Note that the first table has been cropped, since the output is wider than the page width.

14.1.3. Manually broken paragraphs in table cells

Sometimes it is necessary to not rely on the breaking algorithm when using the `p` specifier, but rather specify the line breaks by hand. In this case it is easiest to use a `\parbox`:

```
\begin{tabular}{cc}
  boring cell content & \parbox[t]{5cm}{rather long par\\new par}
\end{tabular}
```

14.1.4. Space between columns

To tweak the space between columns (LaTeX will by default choose very tight columns), one can alter the column separation: `\setlength{\tabcolsep}{5pt}`. The default value is 6pt.

14.1.5. Space between rows

Re-define the `\arraystretch` command to set the space between rows:

```
\renewcommand{\arraystretch}{1.5}
```

Default value is 1.0.

An alternative way to adjust the rule spacing is to add `\noalign{\smallskip}` before or after the `\hline` and `\cline{i-j}` commands:

```
\begin{tabular}{ l l r }
  \hline\noalign{\smallskip}
  \multicolumn{2}{c}{Item} \\
  \cline{1-2}\noalign{\smallskip}
  Animal & Description & Price (\$) \\
  \noalign{\smallskip}\hline\noalign{\smallskip}
  Gnat  & per gram & 13.65 \\
        & each     &  0.01 \\
  Gnu   & stuffed  & 92.50 \\
  Emu   & stuffed  & 33.33 \\
  Armadillo & frozen & 8.99 \\
  \noalign{\smallskip}\hline
\end{tabular}
```

You may also specify the skip after a line explicitly using glue after the line terminator

```
\begin{tabular}{ll}
\hline
Mineral & Color \\[1cm]
Ruby & red \\
Sapphire & blue \\
\hline
\end{tabular}
```

14.1.6. Other environments inside tables

If you use some LaTeX environments inside table cells, like `verbatim` or `enumerate`:

```
\begin{tabular}{c c}
        \hline
        \begin{verbatim}
        code
        \end{verbatim}
        & description
         \\ \hline
\end{tabular}
```

you might encounter errors similar to

```
! LaTeX Error: Something's wrong--perhaps a missing \item.
```

To solve this problem, change column specifier[6] to ”paragraph” (`p`, `m` or `b`).

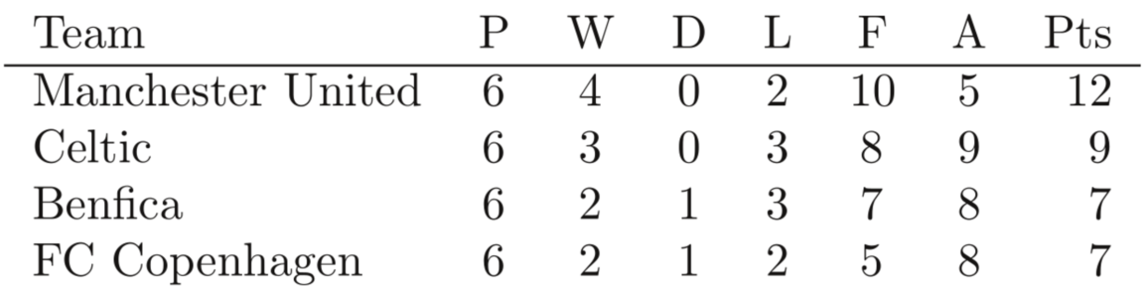
```
\begin{tabular}{m{5cm} c}
```

14.1.7. Defining multiple columns

It is possible to define many identical columns at once using the `*{''num''}{''str''}` syntax. This is particularly useful when your table has many columns.

Here is a table with six centered columns flanked by a single column on each side:

```
\begin{tabular}{l*{6}{c}r}
Team              & P & W & D & L & F  & A & Pts \\
\hline
Manchester United & 6 & 4 & 0 & 2 & 10 & 5 & 12  \\
Celtic            & 6 & 3 & 0 & 3 &  8 & 9 &  9  \\
Benfica           & 6 & 2 & 1 & 3 &  7 & 8 &  7  \\
FC Copenhagen     & 6 & 2 & 1 & 3 &  5 & 8 &  7  \\
\end{tabular}
```



Team	P	W	D	L	F	A	Pts
Manchester United	6	4	0	2	10	5	12
Celtic	6	3	0	3	8	9	9
Benfica	6	2	1	3	7	8	7
FC Copenhagen	6	2	1	2	5	8	7

Figure 39

14.1.8. Column specification using *>{\cmd}* and *<{\cmd}*

The column specification can be altered using the `array` package. This is done in the argument of the tabular environment using `>{\command}` for commands executed right *before* each column element and `<{\command}` for commands to be executed right *after* each column element. As an example: to get a column in math mode enter: `\begin{tabular}{>{$}c<{$}}`. Another example is changing the font: `\begin{tabular}{>{\small}c}` to print the column in a small font.

The argument of the `>` and `<` specifications must be correctly balanced when it comes to `{` and `}` characters. This means that `>{\bfseries}` is valid, while `>{\textbf}` will not work and `>{\textbf{}` is not valid. If there is the need to use the text of the table as an argument

6 Chapter 14.6 on page 173

(for instance, using the `\textbf` to produce bold text), one should use the `\bgroup` and `\egroup` commands: `>{\textbf\bgroup}c<{\egroup}` produces the intended effect. This works only for some basic LaTeX commands. For other commands, such as `\underline` to underline text, it is necessary to temporarily store the column text in a box using `lrbox`. First, you must define such a box with `\newsavebox{\boxname}` and then you can define:

```
>{\begin{lrbox}{\boxname} }%
l%
<{\end{lrbox}%
  \underline{\unhbox\boxname} }%
 }
```

This stores the text in a box and afterwards, takes the text out of the box with `\unhbox` (this destroys the box, if the box is needed again one should use `\unhcopy` instead) and passing it to `\underline`. (For LaTeX2e, you may want to use `\usebox{\boxname}` instead of `\unhbox\boxname`.)

This same trick done with `\raisebox` instead of `\underline` can force all lines in a table to have equal height, instead of the natural varying height that can occur when e.g. math terms or superscripts occur in the text.

Here is an example showing the use of both `p{...}` and `>{\centering}` :

```
\begin{tabular}{>{\centering}p{3.5cm}<{\centering}p{3.5cm} }
Geometry  & Algebra
\tabularnewline
\hline
 Points & Addition
\tabularnewline
 Spheres & Multiplication
\end{tabular}
```

Note the use of `\tabularnewline` instead of `\\` to avoid a `Misplaced \noalign` error.

14.1.9. @-expressions

The column separator can be specified with the `@{...}` construct.

It typically takes some text as its argument, and when appended to a column, it will automatically insert that text into each cell in that column before the actual data for that cell. This command kills the inter-column space and replaces it with whatever is between the curly braces. To add space, use `@{\hspace{''width''}}`.

Admittedly, this is not that clear, and so will require a few examples to clarify. Sometimes, it is desirable in scientific tables to have the numbers aligned on the decimal point. This can be achieved by doing the following:

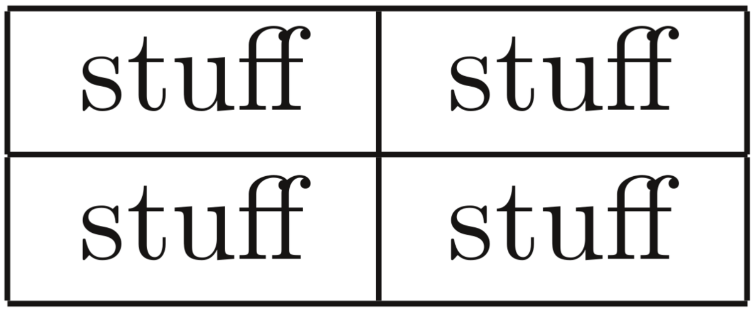
```
 \begin{tabular}{r@{.}l}
   3   & 14159 \\
   16  & 2     \\
   123 & 456   \\
 \end{tabular}
```

3.14159
16.2
123.456

The space-suppressing qualities of the @-expression actually make it quite useful for manipulating the horizontal spacing between columns. Given a basic table, and varying the column descriptions:

```
\begin{tabular}{ ll }
  \hline
  stuff & stuff \\ \hline
  stuff & stuff \\
  \hline
\end{tabular}
```

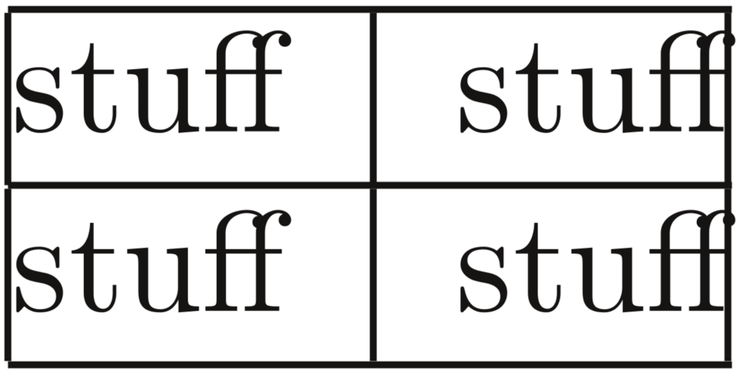
`{|l|l|}`



stuff	stuff
stuff	stuff

Figure 40

`{|@{}l|l@{}|}`



stuff	stuff
stuff	stuff

Figure 41

`{|@{}l@{}|l@{}|}`

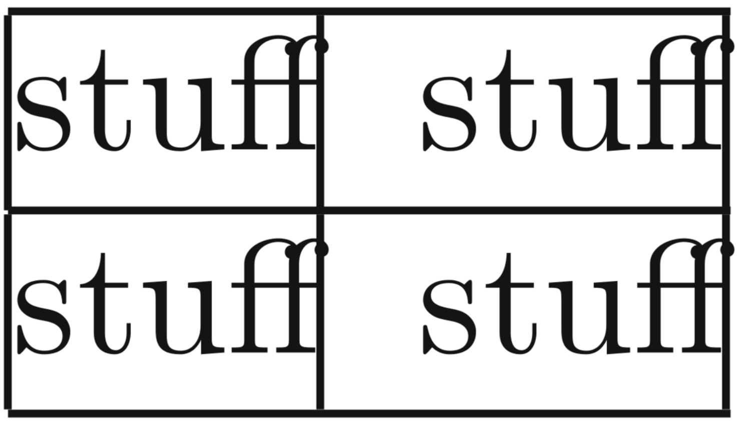


Figure 42

`{|@{}l@{}|@{}l@{}|}`

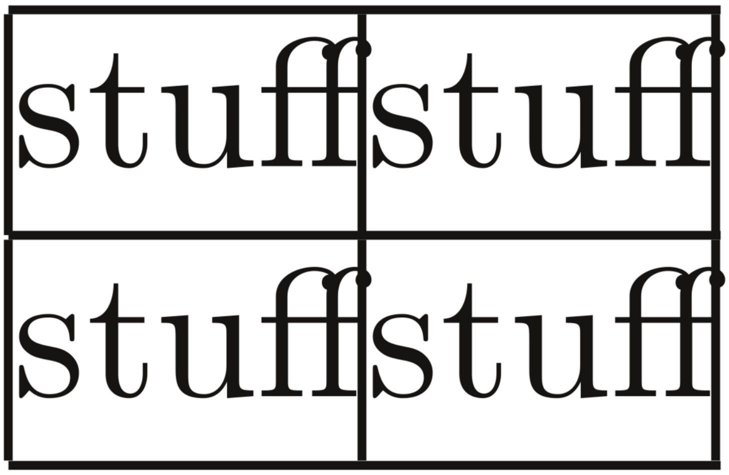


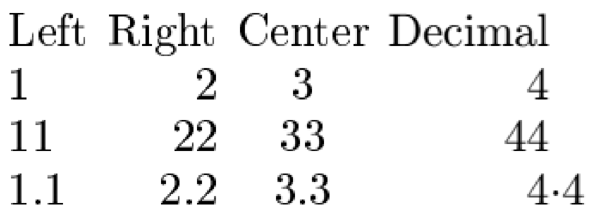
Figure 43

14.1.10. Aligning columns at decimal points using dcolumn

Instead of using @-expressions to build columns of decimals aligned to the decimal point (or equivalent symbol), it is possible to center a column on the decimal separator using the `dcolumn` package, which provides a new column specifier for floating point data. See the dcolumn package documentation[7] for more information, but a simple way to use `dcolumn` is as follows.

7 `http://anorien.csc.warwick.ac.uk/mirrors/CTAN/macros/latex/required/tools/dcolumn.pdf`

```
\usepackage{dcolumn}
\ldots
\newcolumntype{d}[1]{D{.}{\cdot}{#1} }
%the argument for d specifies the maximum number of decimal places
\begin{tabular}{l r c d{1} }
Left&Right&Center&\mathrm{Decimal}\\
1&2&3&4\\
11&22&33&44\\
1.1&2.2&3.3&4.4\\
\end{tabular}
```



Left	Right	Center	Decimal
1	2	3	4
11	22	33	44
1.1	2.2	3.3	4·4

Figure 44

A negative argument provided for the number of decimal places in the new column type allows unlimited decimal places, but may result in rather wide columns. Rounding is not applied, so the data to be tabulated should be adjusted to the number of decimal places specified. Note that a decimal aligned column is typeset in math mode, hence the use of \mathrm for the column heading in the example above. Also, text in a decimal aligned column (for example the header) will be right-aligned before the decimal separator (assuming there's no decimal separator in the text). While this may be fine for very short text, or numeric column headings, it looks cumbersome in the example above. A solution to this is to use the `\multicolumn` command described below, specifying a single column and its alignment. For example to center the header *Decimal* over its column in the above example, the first line of the table itself would be `Left&Right&Center&\multicolumn{1}{c}{Decimal}\\`

Bold text and dcolumn

To draw attention to particular entries in a table, it may be nice to use bold text. Ordinarily this is easy, but as dcolumn needs to *see* the decimal point it is rather harder to do. In addition, the usual bold characters are wider than their normal counterparts, meaning that although the decimals may align nicely, the figures (for more than 2--3 digits on one side of the decimal point) will be visibly misaligned. It is however possible to use normal width bold characters and define a new bold column type, as shown below.[8]

8 Decimals in table don't align with dcolumn when bolded[9]. Stackexchange. Retrieved

```
\usepackage{dcolumn}
%here we're setting up a version of the math fonts with normal x-width
\DeclareMathVersion{nxbold}
\SetSymbolFont{operators}{nxbold}{OT1}{cmr} {b}{n}
\SetSymbolFont{letters}  {nxbold}{OML}{cmm} {b}{it}
\SetSymbolFont{symbols}  {nxbold}{OMS}{cmsy}{b}{n}

\begin{document}
\makeatletter
\newcolumntype{d}{D{.}{.}{-1} } %decimal column as before
%wide bold decimal column
\newcolumntype{B}[3]{>{\boldmath\DC@{#1}{#2}{#3} }c<{\DC@end} }
%normal width bold decimal column
\newcolumntype{Z}[3]{>{\mathversion{nxbold}\DC@{#1}{#2}{#3} }c<{\DC@end} }
\makeatother
\begin{tabular}{l l d}
    Type &M & \multicolumn{1}{c}{N} \\
    Normal & 1 & 22222.222 \\
    Bold (standard)&10 & \multicolumn{1}{B{.}{.}{-1} }{22222.222}\\
    Bold (nxbold)&100 & \multicolumn{1}{Z{.}{.}{-1} }{22222.222}\\
\end{tabular}
\end{document}
```

Type	M	N
Normal	1	22222.222
Bold (standard)	10	**22222.222**
Bold (nxbold)	100	**22222.222**

Figure 45

14.2. Row specification

It might be convenient to apply the same command over every cell of a row, just as for column. Unfortunately the `tabular` environment cannot do that by default. We will need `tabu` instead, which provides the `\rowfont` option.

```
\begin{tabu}{XX}
\rowfont{\bfseries\itshape\large} Header1 & Header2 \\
\hline
Cell2 & Cell2
\end{tabu}
```

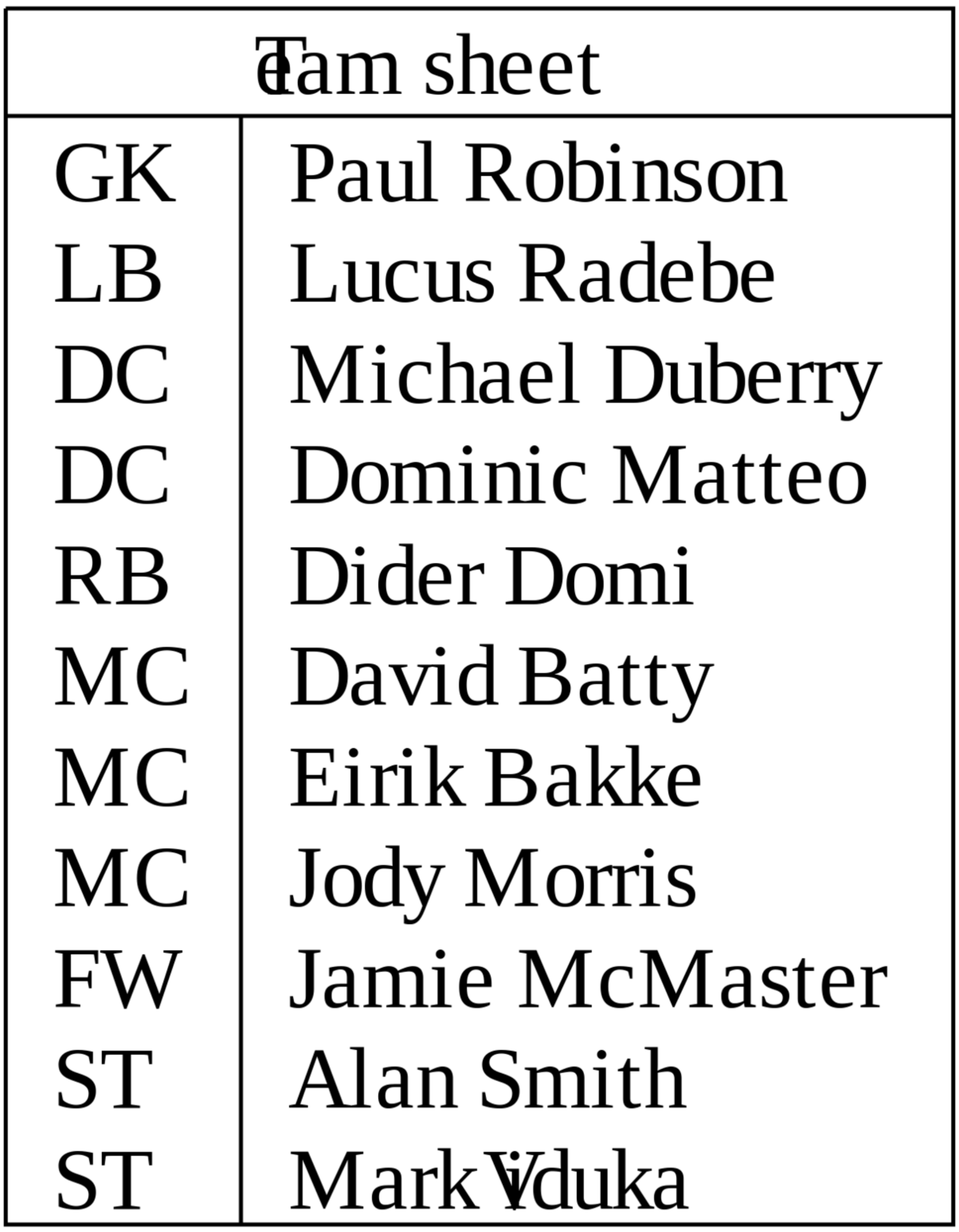
14.3. Spanning

To complete this tutorial, we take a quick look at how to generate slightly more complex tables. Unsurprisingly, the commands necessary have to be embedded within the table data itself.

14.3.1. Rows spanning multiple columns

The command for this looks like this: `\multicolumn{'num_cols'}{'alignment'}{'contents'}`. `num_cols` is the number of subsequent columns to merge; `alignment` is either `l`, `c`, `r`, or to have text wrapping specify a width `p{5.0cm}` . And `contents` is simply the actual data you want to be contained within that cell. A simple example:

```
\begin{tabular}{ ll }
  \hline
  \multicolumn{2}{Team sheet} \\
  \hline
  GK & Paul Robinson \\
  LB & Lucus Radebe \\
  DC & Michael Duberry \\
  DC & Dominic Matteo \\
  RB & Dider Domi \\
  MC & David Batty \\
  MC & Eirik Bakke \\
  MC & Jody Morris \\
  FW & Jamie McMaster \\
  ST & Alan Smith \\
  ST & Mark Viduka \\
  \hline
\end{tabular}
```



Team sheet	
GK	Paul Robinson
LB	Lucus Radebe
DC	Michael Duberry
DC	Dominic Matteo
RB	Dider Domi
MC	David Batty
MC	Eirik Bakke
MC	Jody Morris
FW	Jamie McMaster
ST	Alan Smith
ST	Mark Viduka

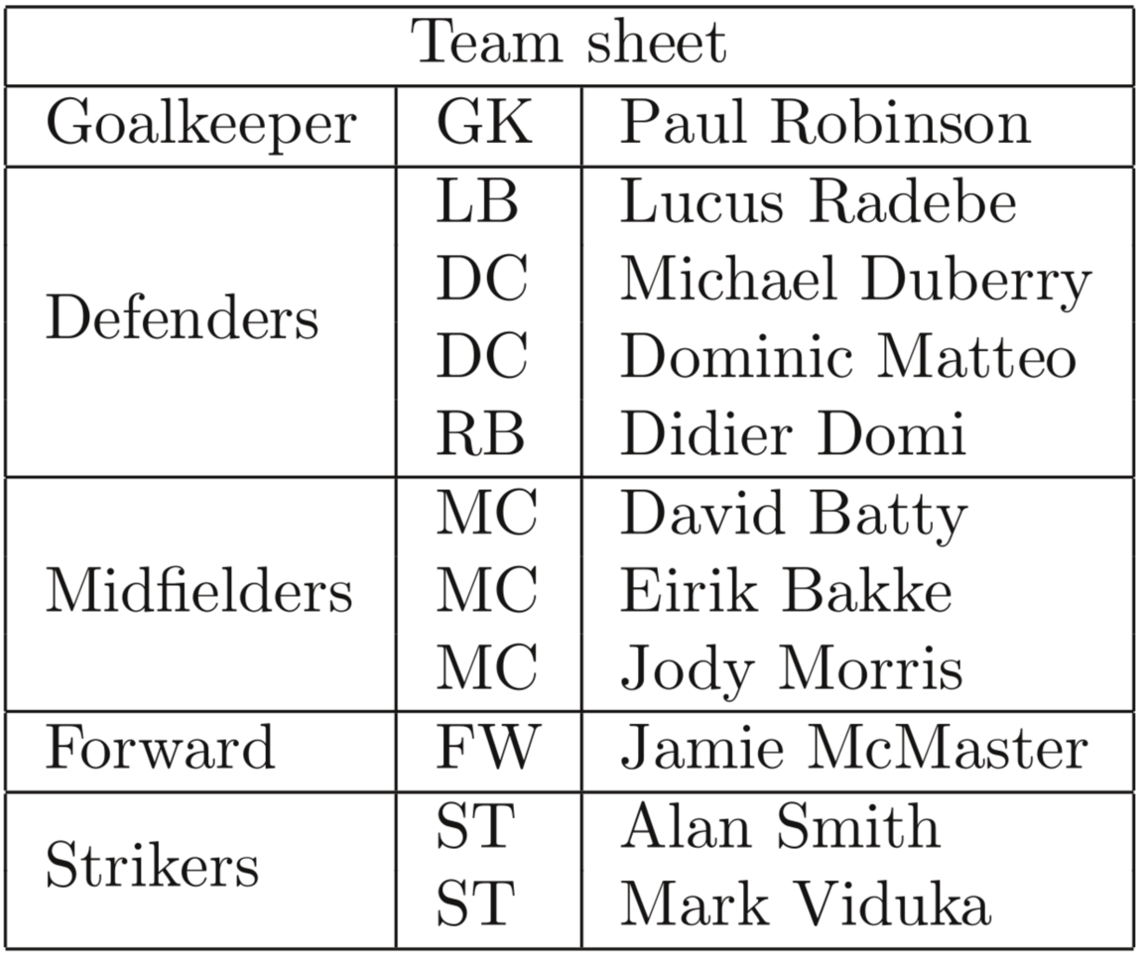
Figure 46

14.3.2. Columns spanning multiple rows

The first thing you need to do is add `\usepackage{multirow}` to the preamble[10]. This then provides the command needed for spanning rows: `\multirow{''num_rows''}{''width''}{''contents''}`. The arguments are pretty simple to deduce (`*` for the *width* means the content's natural width).

10 Package multirow on CTAN ^{`http://www.ctan.org/tex-archive/macros/latex/contrib/multirow/`}

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

\begin{tabular}{ lll }
\hline
\multicolumn{3}{ c }{Team sheet} \\
\hline
Goalkeeper & GK & Paul Robinson \\ \hline
\multirow{4}{*}{Defenders} & LB & Lucus Radebe \\
 & DC & Michael Duburry \\
 & DC & Dominic Matteo \\
 & RB & Didier Domi \\ \hline
\multirow{3}{*}{Midfielders} & MC & David Batty \\
 & MC & Eirik Bakke \\
 & MC & Jody Morris \\ \hline
Forward & FW & Jamie McMaster \\ \hline
\multirow{2}{*}{Strikers} & ST & Alan Smith \\
 & ST & Mark Viduka \\
\hline
\end{tabular}
```



Team sheet		
Goalkeeper	GK	Paul Robinson
Defenders	LB	Lucus Radebe
	DC	Michael Duberry
	DC	Dominic Matteo
	RB	Didier Domi
Midfielders	MC	David Batty
	MC	Eirik Bakke
	MC	Jody Morris
Forward	FW	Jamie McMaster
Strikers	ST	Alan Smith
	ST	Mark Viduka

Figure 47

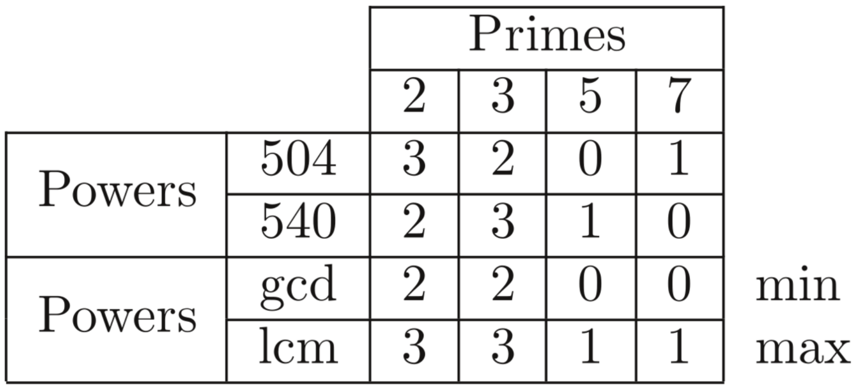
The main thing to note when using `\multirow` is that a blank entry must be inserted for each appropriate cell in each subsequent row to be spanned.

If there is no data for a cell, just don't type anything, but you still need the "&" separating it from the next column's data. The astute reader will already have deduced that for a table of n columns, there must always be $n-1$ ampersands in each row (unless `\multicolumn` is also used).

14.3.3. Spanning in both directions simultaneously

Here is a nontrivial example of how to use spanning in both directions simultaneously and have the borders of the cells drawn correctly:

```
\usepackage{multirow}

\begin{tabular}{cccccl}
\cline{3-6}
& & \multicolumn{4}{ c }{Primes} \\ \cline{3-6}
& & 2 & 3 & 5 & 7 \\ \cline{1-6}
\multicolumn{1}{ c  }{\multirow{2}{*}{Powers} } &
\multicolumn{1}{ c }{504} & 3 & 2 & 0 & 1 &     \\ \cline{2-6}
\multicolumn{1}{ c  }{}                         &
\multicolumn{1}{ c }{540} & 2 & 3 & 1 & 0 &     \\ \cline{1-6}
\multicolumn{1}{ c  }{\multirow{2}{*}{Powers} } &
\multicolumn{1}{ c }{gcd} & 2 & 2 & 0 & 0 & min \\ \cline{2-6}
\multicolumn{1}{ c  }{}                         &
\multicolumn{1}{ c }{lcm} & 3 & 3 & 1 & 1 & max \\ \cline{1-6}
\end{tabular}
```



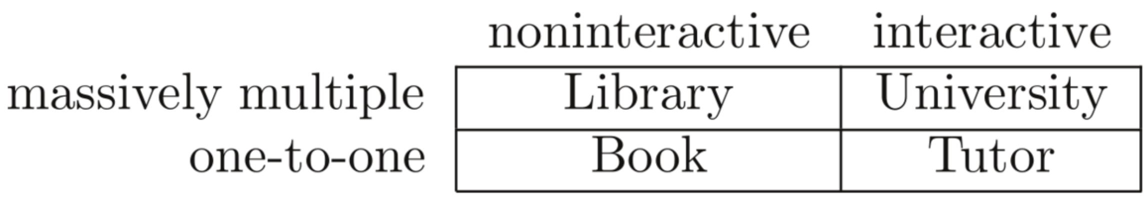
		Primes				
		2	3	5	7	
Powers	504	3	2	0	1	
	540	2	3	1	0	
Powers	gcd	2	2	0	0	min
	lcm	3	3	1	1	max

Figure 48

The command `\multicolumn{1}{` is just used to draw vertical borders both on the left and on the right of the cell. Even when combined with `\multirow{2}{*}{...}`, it still draws vertical borders that only span the first row. To compensate for that, we add `\multicolumn{1}{` in the following rows spanned by the multirow. Note that we cannot just use `\hline` to draw horizontal lines, since we do not want the line to be drawn over the text that spans several rows. Instead we use the command `\cline{2-6}` and opt out the first column that contains the text "Powers".

Here is another example exploiting the same ideas to make the familiar and popular ”2x2” or double dichotomy:

```
\begin{tabular}{ rcc }
\multicolumn{1}{r}{}
 &  \multicolumn{1}{c}{noninteractive}
 & \multicolumn{1}{c}{interactive} \\
\cline{2-3}
massively multiple & Library & University \\
\cline{2-3}
one-to-one & Book & Tutor \\
\cline{2-3}
\end{tabular}
```



	noninteractive	interactive
massively multiple	Library	University
one-to-one	Book	Tutor

Figure 49

14.4. Controlling table size

14.4.1. Resize tables

The `graphicx` packages features the command `\resizebox{width}{height}{object}` which can be used with `tabular` to specify the height and width of a table. The following example shows how to resize a table to 8cm width while maintaining the original width/height ratio.

```
\usepackage{graphicx}
% ...

\resizebox{8cm}{!} {
  \begin{tabular}...
  \end{tabular}
}
```

Resizing table including the caption

```
\begin{table}[h]
\resizebox{1.4\textwidth}{!}{\begin{minipage}{\textwidth}
\begin{tabular}{rcc}
& \multicolumn{1}{c}{noninteractive}
& \multicolumn{1}{c}{interactive} \\
\cline{2-3}
massively multiple & Library & University \\
\cline{2-3}
one-to-one & Book & Tutor \\
\cline{2-3}
\end{tabular}
\caption[Table caption text]{Table taken from \cite[p.10]{refid} }
\label{table:name}
\end{minipage} }
\end{table}
```

Alternatively you can use `\scalebox{ratio}{object}` in the same way but with ratios rather than fixed sizes:

```
\usepackage{graphicx}
% ...

\scalebox{0.7}{
  \begin{tabular}...
  \end{tabular}
}
```

14.4.2. Changing font size

A table can be globally switched to a different font size by simply adding the desired size command (here: `\footnotesize`) in the table scope, which may be after the `\begin{table}` statement if you use floats, otherwise you need to add a group delimiter.

```
{\footnotesize
  \begin{tabular} r  r  c  c  c }
      % ...
  \end{tabular}
}
```

```
\begin{table}[h]\footnotesize
  \caption{Performance at peak F-measure}
  \begin{tabular} r  r  c  c  c }
      % ...
  \end{tabular}
\end{table}
```

Alternatively, you can change the default font for all the tables in your document by placing the following code in the preamble:

```
\let\oldtabular\tabular
\renewcommand{\tabular}{\footnotesize\oldtabular}
```

See Fonts[11] for named font sizes. The table caption font size is not affected. To control the caption font size, see Caption Styles[12].

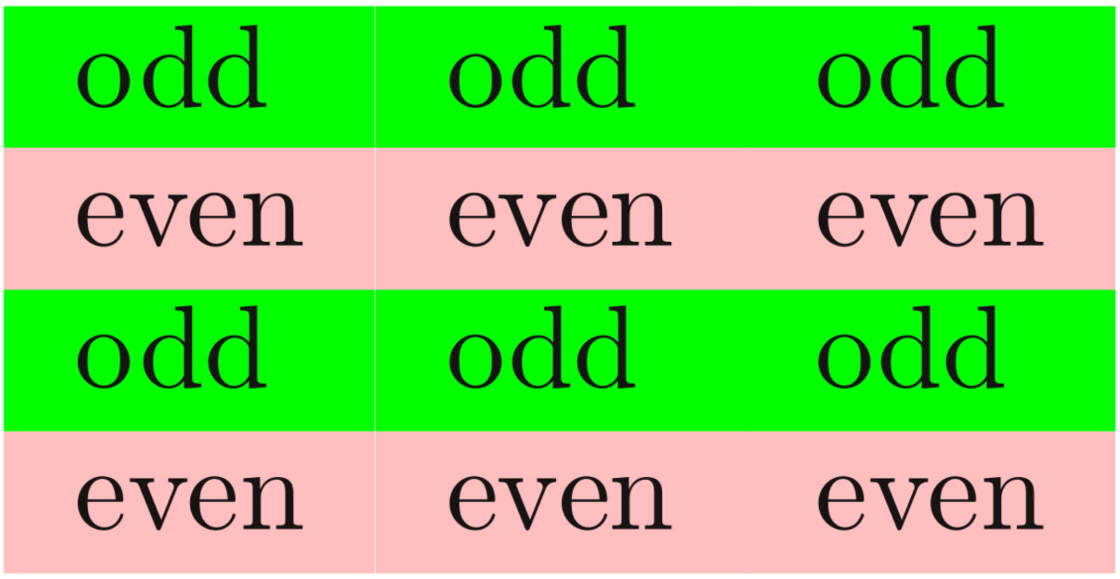
14.5. Colors

14.5.1. Alternate row colors in tables

The `xcolor` package provides the necessary commands to produce tables with alternate row colors, when loaded with the `table` option. The command `\rowcolors{<''starting row''>}{<''odd color''>}{<''even color''>}` has to be specified right before the `tabular` environment starts.

11 Chapter 9.6.2 on page 102
12 Chapter 18.9.1 on page 246

```
\documentclass{article}

\usepackage[table]{xcolor}

\begin{document}

\begin{center}
\rowcolors{1}{green}{pink}

\begin{tabular}{lll}
odd           & odd           & odd \\
even           & even           & even\\
odd           & odd           & odd \\
even           & even           & even\\
\end{tabular}
\end{center}

\end{document}
```



odd	odd	odd
even	even	even
odd	odd	odd
even	even	even

Figure 50

The command `\hiderowcolors` is available to deactivate highlighting from a specified row until the end of the table. Highlighting can be reactivated within the table via the `\showrowcolors` command. If while using these commands you experience ”misplaced \noalign errors” then use the commands at the very beginning or end of a row in your tabular.

```
\hiderowcolors odd & odd & odd \\
```

or

```
odd & odd & odd \\ \showrowcolors
```

14.5.2. Colors of individual cells

As above this uses the `xcolor` package.

```
% Include this somewhere in your document
\usepackage[table]{xcolor}

% Enter this in the cell you wish to color a light grey.
% NB: the word 'gray' here denotes the grayscale color scheme, not the color
 grey. '0.9' denotes how dark the grey is.
\cellcolor[gray]{0.9}
% The following will color the cell red.
\cellcolor{red}
```

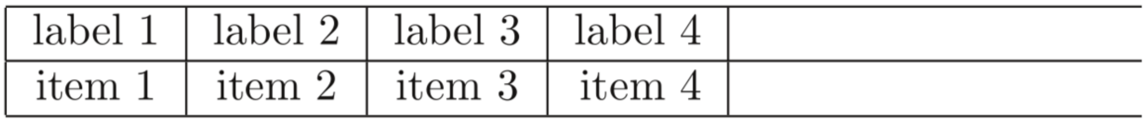
14.6. Width and stretching

We keep providing documentation for `tabular*` and `tabularx` although they are completely eclipsed by the much more powerful and flexible `tabu` environment. Actually `tabu` is greatly inspired by those environments, so it may be worth it to have an idea how they work, particularly for `tabularx`.

14.6.1. The *tabular** environment

This is basically a slight extension on the original tabular version, although it requires an extra argument (before the column descriptions) to specify the preferred width of the table.

```
 \begin{tabular*}{0.75\textwidth}{  c  c  c  r  }
   \hline
   label 1 & label 2 & label 3 & label 4 \\
   \hline
   item 1  & item 2  & item 3  & item 4  \\
   \hline
 \end{tabular*}
```



label 1	label 2	label 3	label 4	
item 1	item 2	item 3	item 4	

Figure 51

However, that may not look quite as intended. The columns are still at their natural width (just wide enough to fit their contents) while the rows are as wide as the table width specified. If you do not like this default, you must also explicitly insert extra column space. LaTeX has *rubber lengths* , which, unlike others, are not fixed. LaTeX can dynamically decide how long the lengths should be. So, an example of this is the following.

```
\begin{tabular*}{0.75\textwidth}{@{\extracolsep{\fill} }  c  c  c  r  }
  \hline
  label 1 & label 2 & label 3 & label 4 \\
  \hline
  item 1  & item 2  & item 3  & item 4  \\
  \hline
\end{tabular*}
```

label 1	label 2	label 3	label 4
item 1	item 2	item 3	item 4

Figure 52

You will notice the `@{...}` construct added at the beginning of the column description. Within it is the `\extracolsep` command, which requires a width. A fixed width could have been used. However, by using a rubber length, such as `\fill`, the columns are automatically spaced evenly.

14.6.2. The *tabularx* package

This package provides a table environment called `tabularx`, which is similar to the `tabular*` environment except that it has a new column specifier `X` (in uppercase). The column(s) specified with this specifier will be stretched to make the table as wide as specified, greatly simplifying the creation of tables.

```
\usepackage{tabularx}
% ...

\begin{tabularx}{\textwidth}{ XXXX }
  \hline
  label 1 & label 2 & label 3 & label 4 \\
  \hline
  item 1  & item 2  & item 3  & item 4  \\
  \hline
\end{tabularx}
```

label 1	label 2	label 3	label 4
item 1	item 2	item 3	item 4

Figure 53

The content provided for the boxes is treated as for a `p` column, except that the width is calculated automatically. If you use the package `array`, you may also apply any `>{\cmd}` or `<{\cmd}` command to achieve specific behavior (like `\centering`, or `\raggedright\arraybackslash`) as described previously.

Another option is to use `\newcolumntype` to format selected columns in a different way. It defines a new column specifier, e.g. `R` (in uppercase). In this example, the second and fourth column is adjusted in a different way (`\raggedleft`):

```
\usepackage{tabularx}
% ...

\newcolumntype{R}{>{\raggedleft\arraybackslash}X}%
\begin{tabularx}{\textwidth}{ lRlR }
  \hline
  label 1 & label 2 & label 3 & label 4 \\
  \hline
  item 1  & item 2  & item 3  & item 4  \\
  \hline
\end{tabularx}
```

label 1	label 2	label 3	label 4
item 1	item 2	item 3	item 4

Figure 54

Tabularx with rows spanning multiple columns using `\multicolumn`. The two central columns are posing as one by using the `X@{}` option. Note that the `\multicolumn` width (which in this example is 2) should equal the (in this example 1+1) width of the spanned columns:

```
\usepackage{tabularx}
% ...

\begin{tabularx}{1\textwidth}{ >{\
setlength\hsize{1\hsize}\centering}X>{\setlength\hsize{1\hsize}\raggedleft}X@{}
>{
\setlength\hsize{1\hsize}\raggedright}X>{\setlength\hsize{1\hsize}\centering}X
 }
  \hline
Label 1 & \multicolumn{2}{>{\centering\setlength\hsize{2\hsize} }X}{Label 2} &
 Label 3\tabularnewline
\hline
  123  & 123  & 456  & 123  \tabularnewline
  \hline
  123  & 123  & 456  & 123  \tabularnewline
  \hline
\end{tabularx}
```

Label 1	Label 2	Label 3
123	123456	123
123	123456	123

Figure 55

In a way analogous to how new commands with arguments can be created with \newcommand, new column types with arguments can be created with \newcolumntype as follows:

```
\usepackage{tabularx}
\usepackage[table]{xcolor} %Used to color the last column
% ...

\newcolumntype{L}[1]{>{\hsize=#1\hsize\raggedright\arraybackslash}X}%
\newcolumntype{R}[1]{>{\hsize=#1\hsize\raggedleft\arraybackslash}X}%
\ne
wcolumntype{C}[2]{>{\hsize=#1\hsize\columncolor{#2}\centering\arraybackslash}X}%

\begin{tabularx}{\textwidth}{  L{1}  R{0.5}  R{0.5}  C{2}{gray}  }
  \hline
  label 1 & label 2 & label 3 & label 4 \\
  \hline
  item 1  & item 2  & item 3  & item 4  \\
  \hline
\end{tabularx}
```

where since there are 4 columns, the sum of the \hsize's (1 + 0.5 + 0.5 + 2) must be equal to 4. The default value used by tabularx for \hsize is 1.

14.6.3. The *tabulary* package

`tabulary` [13] is a modified `tabular*` allowing width of columns set for equal heights. `tabulary` allows easy and convenient writing of well balanced tables.

The problem with `tabularx` is that it leaves much blank if your cells are almost empty. Besides, it is not easy to have different column sizes.

`tabulary` tries to balance the column widths so that each column has at least its natural width, without exceeding the maximum length.

```
\usepackage{tabulary}
...

\begin{center}
  \begin{tabulary}{0.7\textwidth}{LCL}
    Short sentences      & \#  & Long sentences
                \\
    \hline
    This is short.       & 173 & This is much looooooonger, because there are
 many more words.  \\
    This is not shorter. & 317 & This is still looooooonger, because there are
 many more words. \\
  \end{tabulary}
\end{center}
```

The first parameter is the maximum width. `tabulary` will try not to exceed it, but it will not stretch to it if there is not enough content, contrary to `tabularx`.

The second parameter is the column disposition. Possible values are those from the `tabular` environment, plus

`L`	left-justified balanced column
`C`	centered balanced column
`R`	right-justified balanced column

13 `http://www.ctan.org/pkg/tabulary`

J	left-right-justified balanced column

These are all capitals.

14.6.4. The *tabu* environment

It works pretty much like `tabularx`.

```
\begin{tabu} to \linewidth {llX[2]lllXl}
% ...
\end{tabu}
```

'`to \linewidth`' specifies the target width. The `X` parameter can have an optional span factor.

14.7. Table across several pages

Long tables are natively supported by LaTeX thanks to the `longtable` environment. Unfortunately this environment does not support stretching (X columns).

The `tabu` packages provides the `longtabu` environment. It has most of the features of `tabu`, with the additional capability to span multiple pages.

LaTeX can do well with long tables: you can specify a header that will repeat on every page, a header for the first page only, and the same for the footer.

```
\begin{longtabu} to \linewidth {lX[2]lXl}

\rowfont\bfseries H1 & H2 & H3 & H4 & H5 \\ \hline
\endhead

\\ \hline
\multicolumn{5}{r}{There is more to come} \\
\endfoot

\\ \hline
\endlastfoot

% Content ...
```

It uses syntax similar to `longtable`, so you should have a look at its documentation if you want to know more.

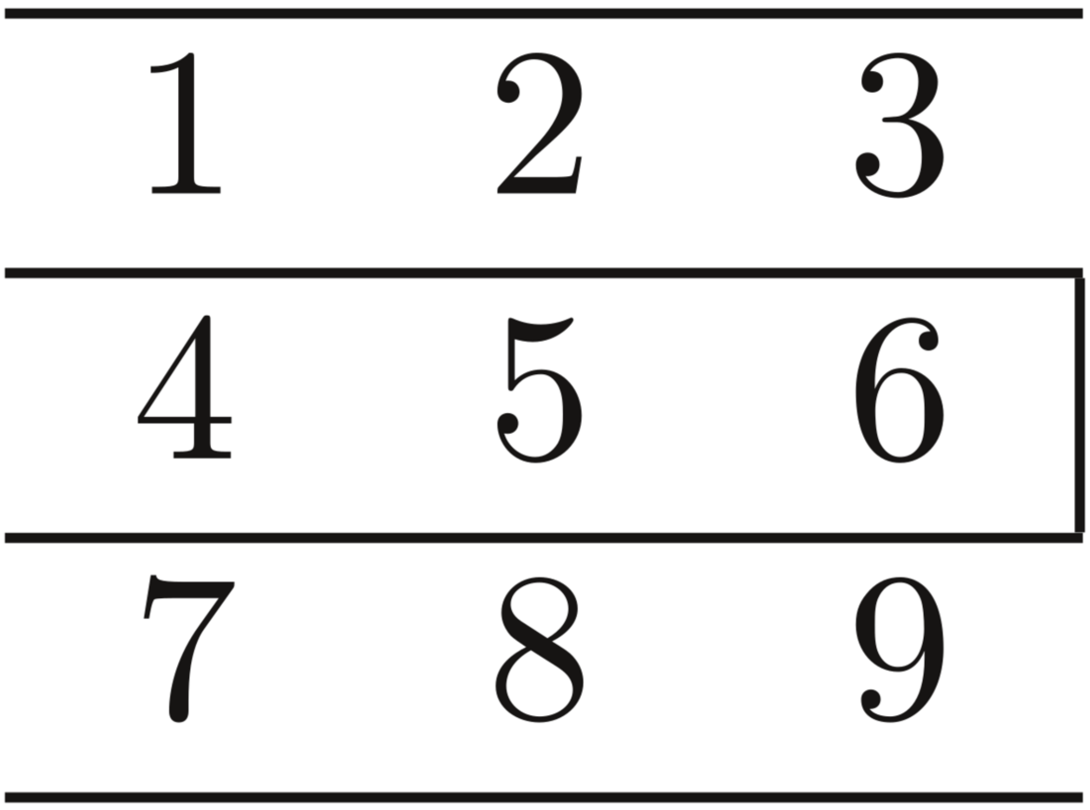
Alternatively you can try one of the following packages `supertabular` [14] or `xtab` [15], an extended and somewhat improved version of `supertabular`.

14 `http://www.ctan.org/pkg/supertabular`
15 `http://www.ctan.org/pkg/xtab`

14.8. Partial vertical lines

Adding a partial vertical line to an individual cell:

```
\begin{tabular}{ l c r }
  \hline
  1 & 2 & 3 \\ \hline
  4 & 5 & \multicolumn{1}{r}{6}  \\ \hline
  7 & 8 & 9 \\ \hline
\end{tabular}
```

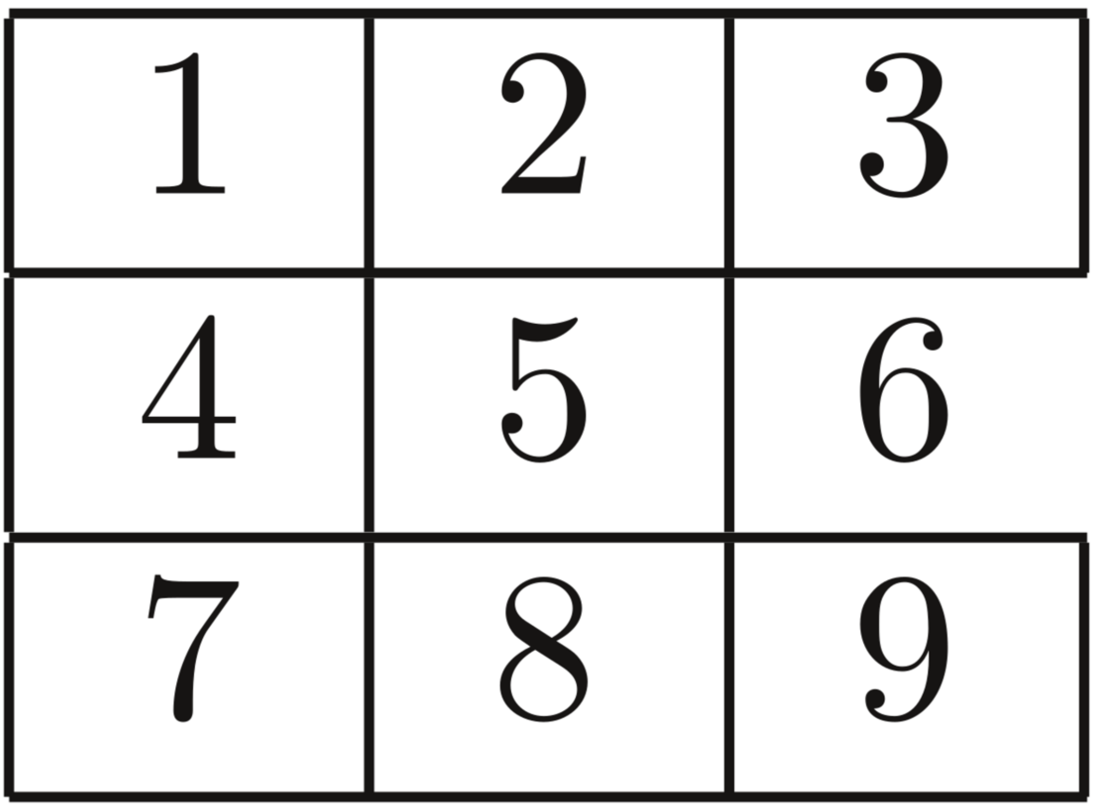


1	2	3
4	5	6
7	8	9

Figure 56

Removing part of a vertical line in a particular cell:

```
\begin{tabular}{  l  c  r  }
  \hline
  1 & 2 & 3 \\ \hline
  4 & 5 & \multicolumn{1}{r}{6} \\ \hline
  7 & 8 & 9 \\ \hline
\end{tabular}
```



1	2	3
4	5	6
7	8	9

Figure 57

14.9. Vertically centered images

Inserting images into a table row will align it at the top of the cell. By using the `array` package this problem can be solved. Defining a new columntype will keep the image vertically centered.

```
\newcolumntype{V}{>{\centering\arraybackslash} m{.4\linewidth} }
```

Or use a parbox to center the image.

```
\parbox[c]{1em}{\includegraphics{image.png} }
```

A raisebox works as well, also allowing to manually fine-tune the alignment with its first parameter.

```
\raisebox{-.5\height}{\includegraphics{image.png} }
```

14.10. Footnotes in tables

The `tabular` environment does not handle footnotes properly. The `longtabular` fixes that.

Instead of using `longtabular` we recommend `tabu` which handles footnotes properly, both in normal and long tables.

14.11. Professional tables

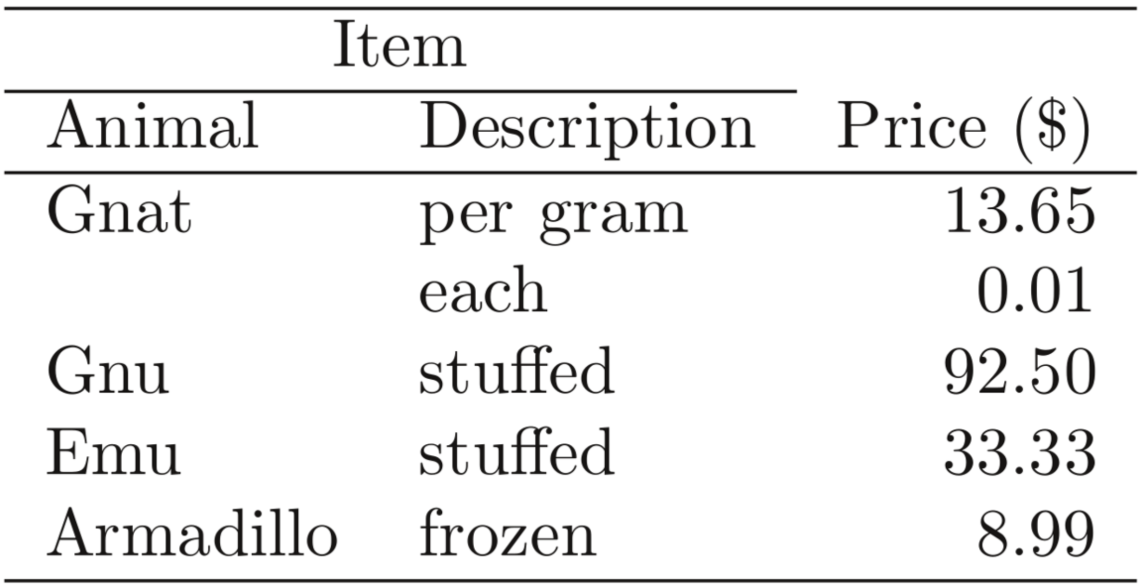
Many professionally typeset books and journals feature simple tables, which have appropriate spacing above and below lines, and almost *never* use vertical rules. Many examples of LaTeX tables (including this Wikibook) showcase the use of vertical rules (using "|"), and double-rules (using `\hline\hline` or "||"), which are regarded as unnecessary and distracting in a professionally published form. The booktabs[16] package is useful for easily providing this professionalism in LaTeX tables, and the documentation[17] also provides guidelines on what constitutes a "good" table.

In brief, the package uses `\toprule` for the uppermost rule (or line), `\midrule` for the rules appearing in the middle of the table (such as under the header), and `\bottomrule` for the lowermost rule. This ensures that the rule weight and spacing are acceptable. In addition, `\cmidrule` can be used for mid-rules that span specified columns. The following example contrasts the use of booktabs and two equivalent normal LaTeX implementations (the second example requires `\usepackage{array}` or `\usepackage{dcolumn}`, and the third example requires `\usepackage{booktabs}` in the preamble).

14.11.1. Normal LaTeX

```
\begin{tabular}{llr}
\hline
\multicolumn{2}{c}{Item} \\
\cline{1-2}
Animal    & Description & Price (\$) \\
\hline
Gnat      & per gram    & 13.65      \\
          & each        & 0.01       \\
Gnu       & stuffed     & 92.50      \\
Emu       & stuffed     & 33.33      \\
Armadillo & frozen      & 8.99       \\
\hline
\end{tabular}
```

16 http://www.ctan.org/tex-archive/macros/latex/contrib/booktabs/
17 http://mirrors.ctan.org/macros/latex/contrib/booktabs/booktabs.pdf



Item		
Animal	Description	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

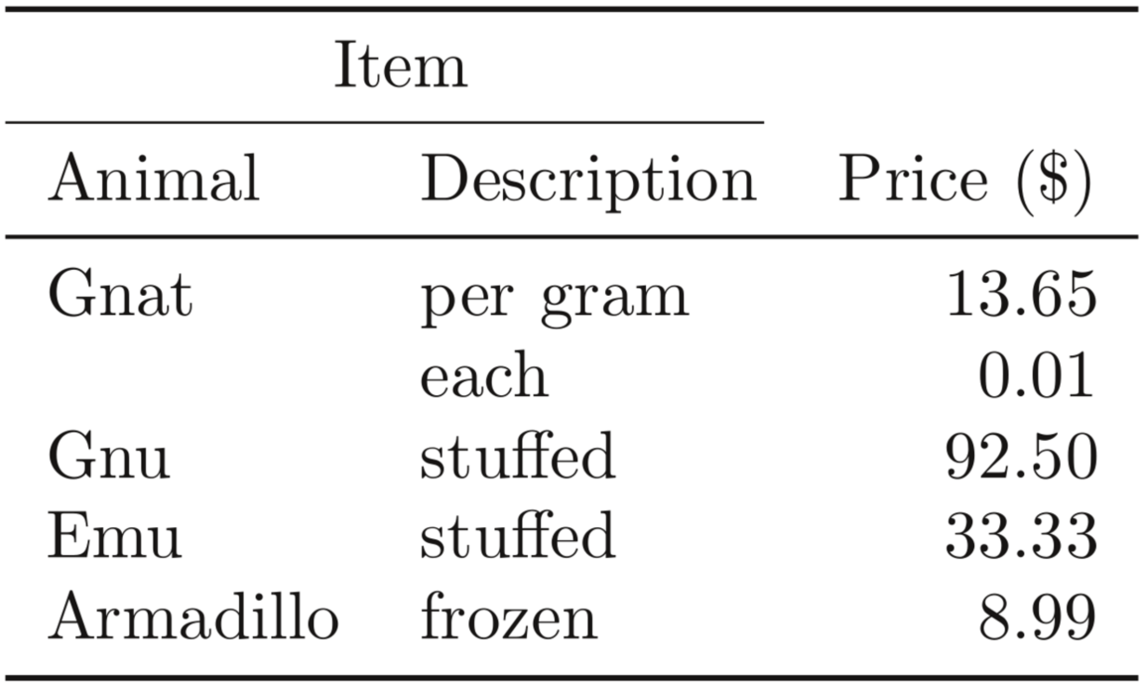
Figure 58

14.11.2. Using array

```
\usepackage{array}
%or \usepackage{dcolumn}
...
\begin{tabular}{llr}
\firsthline
\multicolumn{2}{c}{Item} \\
\cline{1-2}
Animal    & Description & Price (\$) \\
\hline
Gnat      & per gram    & 13.65      \\
          & each        & 0.01       \\
Gnu       & stuffed     & 92.50      \\
Emu       & stuffed     & 33.33      \\
Armadillo & frozen      & 8.99       \\
\lasthline
\end{tabular}
```

14.11.3. Using booktabs

```
\usepackage{booktabs}

...
\begin{tabular}{llr}
\toprule
\multicolumn{2}{c}{Item} \\
\cmidrule(r){1-2}
Animal    & Description & Price (\$) \\
\midrule
Gnat      & per gram    & 13.65      \\
          & each        & 0.01       \\
Gnu       & stuffed     & 92.50      \\
Emu       & stuffed     & 33.33      \\
Armadillo & frozen      & 8.99       \\
\bottomrule
\end{tabular}
```



Item		
Animal	Description	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

Figure 59

Usually the need arises for footnotes under a table (and not at the bottom of the page), with a caption properly spaced above the table. These are addressed by the ctable[18] package. It provides the option of a short caption given to be inserted in the list of tables, instead of the actual caption (which may be quite long and inappropriate for the list of tables). The `ctable` uses the `booktabs` package.

14.12. Sideways tables

Tables can also be put on their side within a document using the `rotating` or the `rotfloat` package. See the Rotations[19] chapter.

14.13. Table with legend

To add a legend to a table the caption[20] package can be used. With the caption package a `\caption*{...}` statement can be added besides the normal `\caption{...}`. Example:

18 `http://www.ctan.org/tex-archive/macros/latex/contrib/ctable/`
19 Chapter 13 on page 151
20 `http://www.ctan.org/tex-archive/macros/latex/contrib/caption/`

```
\begin{table}
  \begin{tabular} r  r  c  c  c }

      ...

  \end{tabular}
  \caption{A normal caption}
  \caption*{
    A legend, even a table can be used
    \begin{tabular}{l l}
      item 1 & explanation 1 \\
    \end{tabular}
  }
\end{table}
```

The normal caption is needed for labels and references.

14.14. The *eqparbox* package

On rare occasions, it might be necessary to stretch every row in a table to the natural width of its longest line, for instance when one has the same text in two languages and wishes to present these next to each other with lines synching up. A tabular environment helps control where lines should break, but cannot justify the text, which leads to ragged right edges. The `eqparbox` package provides the command `\eqmakebox` which is like `\makebox` but instead of a `width` argument, it takes a tag. During compilation it bookkeeps which `\eqmakebox` with a certain tag contains the widest text and can stretch all `\eqmakebox`es with the same tag to that width. Combined with the `array` package, one can define a column specifier that justifies the text in all lines:

```
\newsavebox{\tstretchbox}
\newcolumntype{S}[1]{%
  >{\begin{lrbox}{\tstretchbox} }%
  l%
  <{\end{lrbox}%
  \eqmakebox[#1][s]{\unhcopy\tstretchbox} }%
}
```

See the documentation of the `eqparbox` package for more details.

14.15. Floating with *table*

In WYSIWYG document processors, it is common to put tables in the middle of the text. This is what we have been doing until now. Professional documents, however, often make it a point to print tables on a dedicated page so that they do not disrupt the flow. From the point of view of the source code, one has no idea on which page the current text is going to lie, so it is hardly possible to guess which page may be appropriate for our table. LaTeX can automate this task by abstracting objects such as tables, pictures, etc., and deciding for us where they might fit best. This abstraction is called a *float* . Generally, an object that is floated will appear in the vicinity of its introduction in the source file, but one can choose to control its position also.

To tell LaTeX we want to use our table as a float, we need to place a `tabular` environment in a `table` environment, which is able to float and add a label and caption.

> ⚠ **Warning**
>
> Please understand: you do not *have to* use floating tables. If you want to place your tables where they lie in your source code and you do not need any label, do not use `table` at all! This is a very common misunderstanding among newcomers.

The `table` environment initiates a type of float just as the environment `figure`. In fact, the two bear a lot of similarities (positioning, captions, etc.). More information about floating environments, captions etc. can be found in Floats, Figures and Captions[21].

The environment names may now seem quite confusing. Let's sum it up:

- `tabular` is for the content itself (columns, lines, etc.).
- `table` is for the location of the table on the document, plus caption and label support.

```
\begin{table}[position specifier]
  \centering
  \begin{tabular}l}
    ... your table ...
  \end{tabular}
  \caption{This table shows some data}
  \label{tab:myfirsttable}
\end{table}
```

In the table, we used a label, so now we can refer to it just like any other reference:

```
\ref{tab:myfirsttable}
```

The `table` environment is also useful when you want to have a list of tables at the beginning or end of your document with the command

```
\listoftables
```

The captions show now up in the list of tables, if displayed.

You can set the optional parameter `position specifier` to define the position of the table, where it should be placed. The following characters are all possible placements. Using sequences of it define your "wishlist" to LaTeX.

`h`	where the table is declared (**h** ere)
`t`	at the **t** op of the page
`b`	at the **b** ottom of the page
`p`	on a dedicated **p** age of floats
`!`	override the default float restrictions. E.g., the maximum size allowed of a `b` float is normally quite small; if you want a large one, you need this `!` parameter as well.

21 Chapter 18 on page 231

Default is *tbp* , which means that it is by default placed on the top of the page. If that's not possible, it's placed at the bottom if possible, or finally with other floating environments on an extra page.

You can force LaTeX to use one given position. E.g. *[!h]* forces LaTeX to place it exactly where you place it (Except when it's really impossible, e.g you place a table *here* and this place would be the last line on a page). Again, understand it correctly: it urges LaTeX to put the table at a specific place, but it will not be placed there if LaTeX thinks it will not look great. If you really want to place your table manually, do not use the `table` environment.

Centering the table horizontally works like everything else, using the `\centering` command just after opening the `table` environment, or by enclosing it with a `center` environment.

14.16. Using spreadsheets

For complex or dynamic tables, you may want to use a spreadsheet. You might save lots of time by building tables using specialized software and exporting them in LaTeX format. The following plugins and libraries are available for some popular software:

- calc2latex[22]: for OpenOffice.org Calc spreadsheets,
- excel2latex[23]: for Microsoft Office Excel,
- matrix2latex[24]: for MATLAB,
- matrix2latex[25]: for Python and MATLAB,
- latex-tools[26]: a Ruby library,
- xtable[27]: a library for R,
- org-mode[28]: for Emacs users, org-mode tables can be used inline in LaTeX documents, see `https://www.gnu.org/software/emacs/manual/html_node/org/A-LaTeX-example.html` for a tutorial.
- Emacs align commands[29]: the align commands can clean up a messy LaTeX table.
- Online Table generator for LATeX[30]: An online tool for creating simple tables within the browser. LaTeX format is directly generated as you type.
- Create LaTeX tables online [31]: Online tool.

However, copying the generated source code to your document is not convenient at all. For maximum flexibility, generate the source code to a separate file which you can input from your main document file with the `\input` command. If your speadsheet supports command-line, you can generate your complete document (table included) in one command, using a Makefile for example.

22 `http://calc2latex.sourceforge.net/`
23 `http://www.ctan.org/tex-archive/support/excel2latex/`
24 `http://www.mathworks.com/matlabcentral/fileexchange/4894-matrix2latex`
25 `https://code.google.com/p/matrix2latex/`
26 `http://rubygems.org/gems/latex-tools`
27 `http://cran.r-project.org/web/packages/xtable/index.html`
28 `http://orgmode.org/`
29 `http://emacswiki.org/emacs/AlignCommands`
30 `http://truben.no/latex/table/`
31 `http://www.tablesgenerator.com/`

See Modular Documents[32] for more details.

14.17. Need more complicated features?

Have a look at one of the following packages:

- `hhline` [33]: do whatever you want with horizontal lines
- `array` [34]: gives you more freedom on how to define columns
- `colortbl` [35]: make your table more colorful
- `threeparttable` [36] makes it possible to put footnotes both within the table and its caption
- `arydshln` [37]: creates dashed horizontal and vertical lines
- `ctable` [38]: allows for footnotes under table and properly spaced caption above (incorporates booktabs package)
- `slashbox` [39]: create 2D tables with the first cell containing a description for both axes. Not available in Tex Live 2011 or later.
- `diagbox` [40]: compatible to `slashbox` , come with Tex Live 2011 or later
- `dcolumn` [41]: decimal point alignment of numeric cells
- `rccol` [42]: advanced decimal point alignment of numeric cells with rounding
- `numprint` [43]: print numbers, in the current mode (text or math) in order to use the correct font, with separators, exponent and/or rounded to a given number of digits. tabular(*), array, tabularx, and longtable environments is supported using all features of numprint
- `spreadtab` [44]: spread sheets allowing the use of formulae
- `siunitx` [45]: alignment of tabular entries
- `pgfplotstable` [46]: Loads, rounds, formats and postprocesses numerical tables.

14.18. References

fr:LaTeX/Faire_des_tableaux[47] nl:LaTeX/Tabellen[48] pl:LaTeX/Tabele[49]

32 Chapter 55 on page 607
33 `http://www.ctan.org/pkg/hhline`
34 `http://www.ctan.org/pkg/array`
35 `http://www.ctan.org/pkg/colortbl`
36 `http://ctan.org/tex-archive/macros/latex/contrib/threeparttable`
37 `http://www.ctan.org/pkg/arydshln`
38 `http://www.ctan.org/pkg/ctable`
39 `http://www.ctan.org/pkg/slashbox`
40 `http://mirror.jmu.edu/pub/CTAN/macros/latex/contrib/diagbox/`
41 `http://www.ctan.org/pkg/dcolumn`
42 `http://www.ctan.org/pkg/rccol`
43 `http://www.ctan.org/pkg/numprint`
44 `http://www.ctan.org/pkg/spreadtab`
45 `http://ctan.org/tex-archive/macros/latex/contrib/siunitx`
46 `http://www.ctan.org/tex-archive/graphics/pgf/contrib/pgfplots`
47 `http://fr.wikibooks.org/wiki/LaTeX%2FFaire_des_tableaux`
48 `http://nl.wikibooks.org/wiki/LaTeX%2FTabellen`
49 `http://pl.wikibooks.org/wiki/LaTeX%2FTabele`

15. Title creation

For documents such as basic articles, the output of `\maketitle` is often adequate, but longer documents (such as books and reports) often require more involved formatting. We will detail the process here.

There are several situations where you might want to create a title in a custom format, rather than in the format natively supported by LaTeX. While it is possible to change the output of `\maketitle`, it can be complicated even with minor changes to the title. In such cases it is often better to create the title from scratch, and this section will show you how to accomplish this.

15.1. Standard Title Pages

Many document classes will form a title page for you. One must specify what to fill it with using these commands placed in the top matter[1]:

```
\title{The Triangulation of Titling Data in
       Non-Linear Gaussian Fashion via $\rho$ Series}
\date{October 31, 475}
\author{John Doe\\ Magic Department, Richard Miles University
        \and Richard Row, \LaTeX\ Academy}
```

Commonly the date is excluded from the title page by using `\date{}`. It defaults to `\today` if not in the source file.

To form a title page, use

```
\maketitle
```

This should go after the preceding commands. For most document styles, this will form a separate page, while the `article` document style will place the title on the top of the first page. Note that the `abstract`[2] environment should precede the `\maketitle` command in AMS documents.

Footnotes within the title page can be specified with the `\thanks` command. For example, one may add

```
\author{John Doe\thanks{Funded by NASA Grant \#42}}
```

The `\thanks` command can also be used in the `\title`.

1 Chapter 5.3.1 on page 55
2 Chapter 7.6.5 on page 87

It is dependent on the document class which commands are used in the title page generated by `\maketitle`. For example, the `amsart` uses commands such as `\address`, `\dedicatory`, `\email` and more in the title page while other classes may only use `\title`.

15.2. Custom Title Pages

15.2.1. Create the title

Normally, the benefit of using LaTeX instead of traditional word processing programs is that LaTeX frees you to concentrate on content by handling margins, justification, and other typesetting concerns. On the other hand, if you want to write your own title format, it is exactly the opposite: you have to take care of everything — this time LaTeX will do nothing to help you. It can be challenging to create your own title format since LaTeX was not designed to be graphically interactive in the adjustment of layout. The process is similar to working with raw HTML with the added step that each time you want to see how your changes look, you have to re-compile the source. While this may seem like a major inconvenience, the benefit is that once the customized title format has been written, it serves as a template for all other documents that would use the title format you have just made. In other words, once you have a layout you like, you can use it for any other documents where you would like the same layout without any additional fiddling with layout.

First step: since you'll be working only on the first page of your document and you'll have to compile very often, you don't have to compile the whole document each time, you only need to take a look at the first page. That is why we'll first create a dummy document for preparing the title and then we'll simply include it within the existing big document we are writing. Call the dummy document `test_title.tex` and put the following code in it:

```
\documentclass[pdftex,12pt,a4paper]{report}

\usepackage[pdftex]{graphicx}

\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}

\begin{document}

\input{./title.tex}
\end{document}
```

It is meant to be compiled with `pdflatex` to create a PDF as its output. It is a very basic document, but take care that it has the same settings of the document you are writing, so the output won't change when you include the title in your document. In this case (see the first line) the font size is set to 12pt and the paper size is an A4. The package `graphicx` is included to insert an image in the title. Then a command is defined called `\HRule`; it will just insert a horizontal line whose length is equal to a line of text and whose thickness is 0.5 mm. (Note that you will need to precede the command `\HRule` with `\noindent`, otherwise it will be indented and you will get an `overfull hbox` warning equal to the normal indent of a paragraph's first line.) If you want you can change its settings in the definition. Finally the document starts and it simply includes the `title.tex` file, that must be placed in the same directory of our dummy file `test_title.tex` .

Now create the `title.tex` and write in it:

```
\begin{titlepage}


\end{titlepage}
```

all the things you want to put in the title must be inside the `titlepage` environment. Now if you compile `test_title.tex` you will see a preview of your title in the `test_title.pdf` file. Here is what you need to know to write your title:

Alignment

if you want to center some text just use `\begin{center} ... \end{center}`. If you want to align it differently you can use the environment `flushright` for *right* -alignment and `flushleft` for *left* -alignment.

Images

the command for including images is the following (the example is for a small logo, but you can introduce any image of any size): `\includegraphics[width=0.15\textwidth]{./logo}`. There is no `\begin{figure}` as you usually do because you don't want it to be floating[3], you just want it there where you placed it. When handling it, remember that it is considered like a big box by the TeX engine.

Text size

If you want to change the size of some text just place it within brackets, *{like this}* , and you can use the following commands (in order of size): `\Huge`, `\huge`, `\LARGE`, `\Large`, `\large`, `\small`, `\footnotesize`, `\tiny`. So for example:

```
{\large this text is slightly bigger than normal}, this one is not.
```

`\normalsize` is used to create text at the default size for the document.

New lines

you can force the start of a new line by \\. If you want to add more vertical space, do so by adding the desired space in square brackets after the \\ command. For example, `\\[1cm]` will insert 1 cm of empty space before the new line (which will not be indented). Using multiple \\ commands will produce `Underfull hbox` messages.

Date

you can insert the date of the current day with the command `\today`. If you do not wish to insert any date, keep it blank e.g.`\date{}`

Filling the page

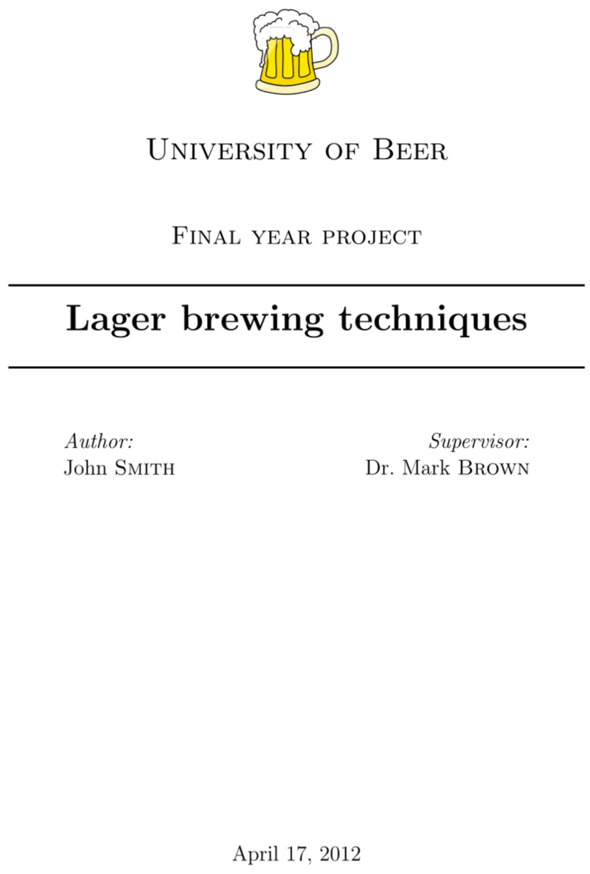
the command `\vfill` as the last item of your content will add empty space until the page is full. If you put it within the page, you will ensure that all the following text will be placed at the bottom of the page.

3 Chapter 18 on page 231

15.2.2. A practical example

All these tips might have made you confused. Then, here is a practical example. Get the `test_title.tex` described above and here is an example of a `title.tex` . On the right you can see the output after you compile `test_title.tex` in PDF:

```
\begin{titlepage}
\begin{center}

% Upper part of the page. The '~' is needed because \\
% only works if a paragraph has started.
\includegraphics[width=0.15\textwidth]{./logo}~\\[1cm]

\textsc{\LARGE University of Beer}\\[1.5cm]

\textsc{\Large Final year project}\\[0.5cm]

% Title
\HRule \\[0.4cm]
{ \huge \bfseries Lager brewing techniques \\[0.4cm] }

\HRule \\[1.5cm]

% Author and supervisor
\noindent
\begin{minipage}[t]{0.4\textwidth}
\begin{flushleft} \large
\emph{Author:}\\
John \textsc{Smith}
\end{flushleft}
\end{minipage}%
\begin{minipage}[t]{0.4\textwidth}
\begin{flushright} \large
\emph{Supervisor:} \\
Dr.~Mark \textsc{Brown}
\end{flushright}
\end{minipage}

\vfill

% Bottom of the page
{\large \today}

\end{center}
\end{titlepage}
```



UNIVERSITY OF BEER

FINAL YEAR PROJECT

Lager brewing techniques

Author:
John SMITH

Supervisor:
Dr. Mark BROWN

April 17, 2012

Figure 60

The picture is from a file called `logo.png` that is in the same directory of both `title.tex` and `test_title.tex` . Since I wanted to insert both the author and supervisor names properly aligned I used a trick: I created two small minipages, one on left and one on the right. Their width is a bit less than half of page width (as you can see, they are exactly 40% of the text width). Within the minipages I have used different alignments. Using `\vfill` I could write the date exactly at the bottom of the page.

As you can see, the code looks ”dirtier” than standard LaTeX source because you have to take care of the output as well. If you start changing font’s output it will get more confused, but you can do it: it’s only for the title and your complicated code will be isolated from all the rest within its own file `title.tex` .

15.2.3. Integrating the title page

Assuming that your title page is now contained in a file named `title.tex` , it must be placed in the same directory as the main document. In order to integrate it, the input command must be used by placing `\input{./title.tex}` at the top of the document. Don't forget to add the commands `\usepackage[pdftex]{graphicx}` and `\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}` in the preamble section as well.

For example, the top section of your document would look like:

```
...
\usepackage[pdftex]{graphicx}

\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}

\begin{document}

\input{./title.tex}
\tableofcontents
...
```

15.3. Packages for custom titles

The `titling` package[4] provides control over the typesetting of the `\maketitle` and `\thanks` commands. The `titlepages` package presents many styles of designs for title pages. Italian users may also want to use the `frontespizio` package[5].

15.4. Notes and References

pt:Latex/Título[6]

4 Titling package webpage in CTAN ^{`http://www.ctan.org/tex-archive/macros/latex/contrib/titling`}

5 Frontespizio package webpage in CTAN ^{`http://www.ctan.org/tex-archive/macros/latex/contrib/frontespizio`}

6 `http://pt.wikibooks.org/wiki/Latex%2FT%C3%ADtulo`

16. Page Layout

LaTeX and the document class will normally take care of page layout issues for you. For submission to an academic publication, this entire topic will be out of your hands, as the publishers want to control the presentation. However, for your own documents, there are some obvious settings that you may wish to change: margins, page orientation and columns, to name but three. The purpose of this tutorial is to show you how to configure your pages.

We will often have to deal with TeX lengths in this chapter. You should have a look at Lengths[1] for comprehensive details on the topic.

16.1. Two-sided documents

Documents can be either one- or two-sided. Articles are by default one-sided, books are two-sided. Two-sided documents differentiate the left (even) and right (odd) pages, whereas one-sided do not. The most notable effect can be seen in page margins. If you want to make the *article class two-sided* , use `\documentclass[twoside]{article}`.

Many commands and variables in LaTeX take this concept into account. They are referred to as *even* and *odd* . For one-sided document, only the *odd* commands and variables will be in effect.

16.2. Page dimensions

A page in LaTeX is defined by many internal parameters. Each parameter corresponds to the length of an element of the page, for example, `\paperheight` is the physical height of the page. Here you can see a diagram showing all the variables defining the page. All sizes are given in TeX points (pt), there are 72.27pt in an inch or 1pt ≈0.3515mm.

1 Chapter 23 on page 283

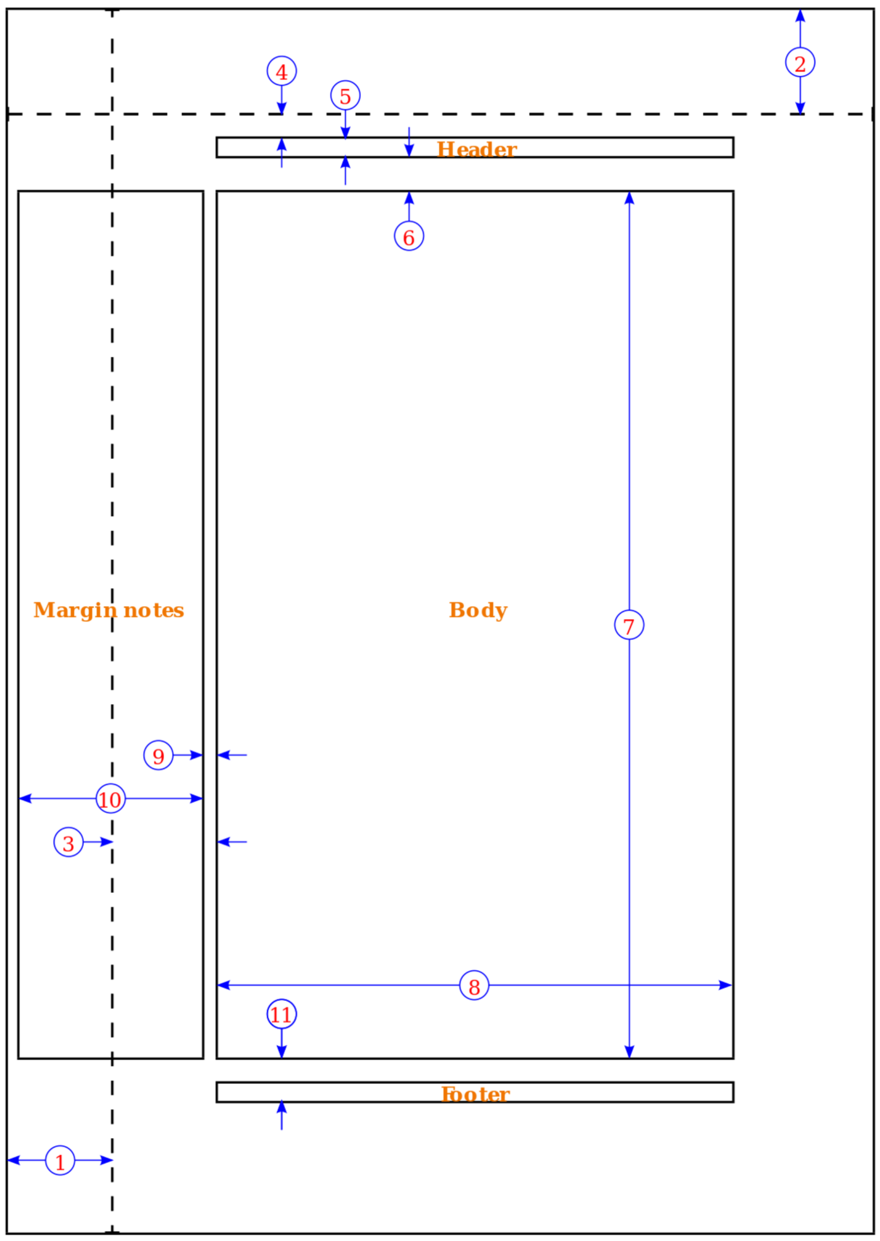


Figure 61

1. one inch + `\hoffset`
2. one inch + `\voffset`
3. `\oddsidemargin` = 31pt
4. `\topmargin` = 20pt
5. `\headheight` = 12pt
6. `\headsep` = 25pt
7. `\textheight` = 592pt
8. `\textwidth` = 390pt
9. `\marginparsep` = 10pt
10. `\marginparwidth` = 35pt

11. `\footskip` = 30pt

- `\marginparpush` = 7pt (not shown)
- `\hoffset` = 0pt
- `\voffset` = 0pt
- `\paperwidth` = 597pt
- `\paperheight` = 845pt

The current details plus the layout shape can be printed from a LaTeX document itself. Use the *layout* package and the command of the same name:

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

To render a frame marking the margins of a document you are currently working on, add

```
\usepackage{showframe}
```

to the document.

16.3. Page size

It will not have been immediately obvious - because it doesn't really cause any serious problems - that the default page size for all standard document classes is *US letter* . This is shorter by 18 mm (about 3/4 inch), and slightly wider by 8 mm (about 1/4 inch), compared to A4 (which is the standard in almost all the rest of the world). While this is not a serious issue (most printers will print the document without any problems), it is possible to specify alternative sizes as class option[2]. For A4 format:

```
\documentclass[a4paper]{article}
```

⚠ **Warning**

Note that the standard LaTeX classes use *US Letter* by default regardless of your TeX distribution configuration. If you have TeX Live configured to use A4 paper, it will be the default only for plainTeX and classes not specifying the paper dimension.

⚠ **Warning**

The a4paper option with the article document class by itself has no effect. It will only affect the page size in connection with some appropriate package, like the `geometry` package or the `hyperref` package.

2 Chapter 5.2.1 on page 52

16.3.1. More size options with *geometry*

One of the most versatile packages for page layout is the `geometry` package. The immediate advantage of this package is that it lets you customize the page size even with classes that do not support the options. For instance, to set the page size, add the following to your preamble:

```
\usepackage[a4paper]{geometry}
```

The `geometry` package has many pre-defined page sizes, like `a4paper`, built in. Others include:

- `a0paper`, `a1paper`, ..., `a6paper`,
- `b0paper`, `b1paper`, ..., `b6paper`,
- `letterpaper`,
- `legalpaper`,
- `executivepaper`.

To explicitly change the paper dimensions using the `geometry` package, the `paperwidth` and `paperheight` options can be used. For example:

```
\usepackage[paperwidth=5.5in, paperheight=8.5in]{geometry}
```

16.3.2. Changing size manually

Use the `\setlength` command to adjust the parameters to the appropriate dimensions. See the Lengths[3] chapter.

- In the preamble, `\paperwidth` and `\paperheight` in all cases.
- After the preamble, `\pdfpagewidth` and `\pdfpageheight` if you are using `pdftex` .

Using the PDF dedicated commands has one immediate advantage: it will let you change the page dimension anywhere in the document.

16.3.3. Page size issues

If you intend to get a PDF in the end, there are basically three ways:

- TeX → PDF

```
pdflatex myfile               # TeX → PDF
```

- TeX → DVI → PDF

```
latex myfile                  # TeX → DVI
dvipdf myfile                 # DVI → PDF
```

- TeX → DVI → PS → PDF

3 Chapter 23 on page 283

```
latex myfile                  # TeX → DVI
dvips myfile -o myfile.ps     # DVI → PS
ps2pdf myfile.ps myfile.pdf   # PS  → PDF
```

Sadly the PDF output page size may not be completely respectful of your settings. Some of these tools do not have the same interpretation of the DVI, PS and PDF specifications, and you may end up with a PDF which has not exactly the right size. Thankfully there is a solution to that: the `\special` command lets the user pass PostScript or PDF parameters, which can be used here to set the page size appropriately.

- For `pdflatex` to work fine, using the package `geometry` usually works.
- For the DVI and PS ways, the safest way to always get the right paper size in the end is to add

```
\documentclass[...,a4paper,...]{...}
\special{papersize=210mm,297mm}
```

to the tex file, and to append the appropriate parameters to the processors used during output generation:

```
dvips -t a4 ...
ps2pdf -sPAPERSIZE=a4 ... # On Windows: ps2pdf -sPAPERSIZE#a4 ... 4
```

If you want US Letter instead, replace `210mm,297mm` by `8.5in,11in` and `a4paper` by `letter`. Also replace `a4` by `letter` in command-line parameters.

16.3.4. Page size for tablets

Those who want to read on tablets or other handheld digital devices need to create documents without the extra whitespace. In order to create PDF documents with optimal handheld viewing, not only must the text field and margins be adjusted, so must the page size. If you are looking for a sensible dimension, consider following the paper size used by the Supreme Court of the United States, 441pt by 666pt (or 6.125 inches by 9.25 inches), which looks great on tablets. You could also use the Supreme Court's text field size of 297 pt by 513 pt, but this is too wide for fonts other than Century Schoolbook, the font required by the Supreme Court.

16.4. Margins

Readers used to perusing typical physical literature are probably wondering why there is so much white space surrounding the text. For example, on A4 paper a document will typically have 44 mm margin widths on the left and right of the page, leaving about 60% of the page width for text. The reason is improved readability. Studies have shown[5][6] that it's

5 `http://webtypography.net/2.1.2`
6 `http://baymard.com/blog/line-length-readability`

easier to read text when there are 60−70 characters per line—and it would seem that 66 is the optimal number. Therefore, the page margins are set to ensure optimal readability, and excessive margin white space is tolerated as a consequence. Sometimes, this white space is left in the inner margin with the assumption that the document will be bound.

If you wish to avoid excessive white space, rather than changing the margins, consider instead using a two-column (or more) layout. This approach is the one usually taken by print magazines because it provides both readable line lengths and good use of the page. Another option for reducing the amount of whitespace on the page without changing the margins is to increase the font size using the `12pt` option to the document class.

If you wish to change the margins of your document, there are many ways to do so:

- One older approach is to use the `fullpage` package for somewhat standardized smaller margins (around an inch), but it creates lines of more than 100 characters per line at with the 10pt default font size (and about 90 if the `12pt` documentclass option is used):

```
\usepackage{fullpage}
```

For even narrower margins, the `fullpage` package has a `cm` option (around 1.5cm), which results in about 120 characters per line at with the 10pt default font size, about double what is considered readable:

```
\usepackage[cm]{fullpage}
```

- A more modern and flexible approach is to use the `geometry` package. This package allows you to specify the 4 margins without needing to remember the particular page dimensions commands. You can enter the measures in centimeters and inches as well. Use `cm` for centimeters and `in` for inches after each value (*e.g.* 1.0in or 2.54cm). Note that by default (*i.e.* without any options) this package already reduces the margins, so for a 'standard layout' you may not need to specify anything. These values are relative to the edge of paper (0in) and go inward it. For example, this command provides more conventional margins, better using the vertical space of the page, without creating the dramatically long lines of the `fullpage` package (if the the `11pt` documentclass option is used, the line lengths are about 88 characters for leter-sized paper and slightly less when using `a4paper`).

```
\usepackage[top=1in, bottom=1.25in, left=1.25in, right=1.25in]{geometry}
```

It can also recreate the behavior of the `fullpage` package using

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

You can combine the margin options with the page size options seen in this paragraph[7].

- You should not use the `a4wide` package for a page with A4 document size with smaller margins. It is obsolete and buggy. Use geometry package instead like this:

```
\usepackage[a4paper,margin=1cm,footskip=.5cm]{geometry}
```

7 Chapter 16.11 on page 209

- Edit individual page dimension variables described above, using the `\addtolength` and `\setlength` commands. See the Lengths[8] chapter. For instance,

```
\setlength{\textwidth}{6.5in}
\addtolength{\voffset}{-5pt}
```

16.4.1. Odd and even margins

Using the geometry package, the options left and right are used for the inside and outside margins respectively. They also have aliases inner and outer. Thus, the easiest way to handle different margins for odd and even pages is to give the twoside option in the document class command and specify the margins as usually.

```
\documentclass[twoside]{report}
\usepackage[inner=4cm,outer=2cm]{geometry} %left=4cm,right=2cm would be
 equivalent
```

This will result in a value of 4cm on all inner margins (left margin for odd number pages and right margin for even pages) and 2cm margin on outer margins.

Setting the same value for the inner and outer for geometry will remove the difference between the margins. Another quick way to eliminate the difference in position between even and odd numbered pages would be setting the values to **evensidemargin** and **oddsidemargin** to the half of odd's default:

```
\setlength{\oddsidemargin}{15.5pt}
\setlength{\evensidemargin}{15.5pt}
```

By default, the value of **evensidemargin** is larger than **oddsidemargin** in the two-sided layout, as one could wish to write notes on the side of the page. The side for the large margin is chosen opposite to the side where pages are joined together.

See the Lengths[9].

16.4.2. Top margin above Chapter

The top margin above a chapter can be changed using the `titlesec` package. Example: `http://www.ctex.org/documents/packages/layout/titlesec.pdf`

```
\usepackage{titlesec}
\titlespacing*{\chapter}{0pt}{-50pt}{20pt}
\titleformat{\chapter}[display]{\normalfont\huge\bfseries}{\chaptertitlename\
 \thechapter}{20pt}{\Huge}
```

The command `\titleformat` must be used when the spacing of a chapter is changed. In case of a section this command can be omitted.

8 Chapter 23 on page 283
9 Chapter 23 on page 283

16.5. Page orientation

When you talk about changing page orientation, it usually means changing to landscape mode, since portrait is the default. We shall introduce two slightly different styles of changing orientation.

16.5.1. Change orientation of the whole document

The first is for when you want all of your document to be in landscape from the very beginning. There are various packages available to achieve this, but the one we prefer is the `geometry` package. All you need to do is call the package, with *landscape* as an option:

```
\usepackage[landscape]{geometry}
```

Although, if you intend to use `geometry` to set your paper size, don't add the `\usepackage` commands twice, simply string all the options together, separating with a comma:

```
\usepackage[a4paper,landscape]{geometry}
```

Using standard LaTeX classes, you can use the same class options:

```
\documentclass[a4paper,landscape]{article}
```

16.5.2. Change orientation of specific part

The second method is for when you are writing a document in portrait, but you have some contents, like a large diagram or table that would be displayed better on a landscape page. However, you still want the consistency of your headers and footers appearing the same place as the other pages.

The `lscape` package is for this very purpose. It supplies a `landscape` environment, and anything inside is basically rotated. No actual page dimensions are changed. This approach is more applicable to books or reports than to typical academic publications. Using `pdflscape` instead of `lscape` when generating a PDF document will make the page appear right side up when viewed: the single page that is in landscape format will be rotated, while the rest will be left in portrait orientation.

Also, to get a table to appear correctly centered on a landscaped page, one must place the `tabular` environment inside a `table` environment, which is itself inside the `landscape` environment. For instance it should look like this:

```
\usepackage{pdflscape}
% ...

\begin{landscape}
\begin{table}
\centering     % optional, probably makes it look better to have it centered on
 the page
\begin{tabular}{....}
% ...
\end{tabular}
```

```
\end{table}
\end{landscape}
```

16.5.3. Change orientation of floating environment

If you use the above code, you will see that the table is inserted where it is in the code. It will not be floated! To fix this you need the package `rotating`. See the Rotations[10] chapter.

16.6. Margins, page size and rotation of a specific page

If you need to rotate the page so that the figure fits, the chances are good that you need to scale the margins and the font size too. Again, the `geometry` package comes in handy for specifying new margins for a single page only.

```
\usepackage{geometry}
\usepackage{pdflscape}
% ...

\newgeometry{margin=1cm}
\begin{landscape}
\thispagestyle{empty} %% Remove header and footer.

\begin{table}
\begin{center}
\footnotesize %% Smaller font size.

\begin{tabular}{....}
% ...
\end{tabular}

\end{center}
\end{table}

\end{landscape}
\restoregeometry
```

Note that order matters!

16.7. Page styles

Page styles in Latex terms refers not to page dimensions, but to the running headers and footers of a document. These headers typically contain document titles, chapter or section numbers/names, and page numbers.

16.7.1. Standard page styles

The possibilities of changing the headers in plain Latex are actually quite limited. There are two commands available: `\pagestyle{''style''}` will apply the specified style to the

10 Chapter 13 on page 151

current and all subsequent pages, and `\thispagestyle{''style''}` will only affect the current page. The possible styles are:

`empty`	Both header and footer are cleared
`plain`	Header is clear, but the footer contains the page number in the center.
`headings`	Footer is blank, header displays information according to document class (e.g., section name) and page number top right.
`myheadings`	Page number is top right, and it is possible to control the rest of the header.

With myheadings, the commands `\markright` (in the standard document classes, `book`, `report` and `article`) and `\markboth` (only in the `book` class) are used to control the headings. The following commands placed at the beginning of an article document will set the header of all pages to contain ”John Smith” top left, ”On page styles” centered and the page number top right:

```
\pagestyle{myheadings}
\markright{John Smith\hfill On page styles\hfill}
```

There are special commands containing details on the running page of the document.

`\thepage`	number of the current page
`\leftmark`	current chapter name printed like ”CHAPTER 3. THIS IS THE CHAPTER TITLE”
`\rightmark`	current section name printed like ”1.6. THIS IS THE SECTION TITLE”
`\chaptername`	the name *chapter* in the current language. If this is English, it will display ”Chapter”
`\thechapter`	current chapter number
`\thesection`	current section number

Note that `\leftmark` and `\rightmark` convert the names to uppercase, whichever was the formatting of the text. If you want them to print the actual name of the chapter without converting it to uppercase use the following command:

```
\renewcommand{\chaptermark}[1]{ \markboth{#1}{} }
\renewcommand{\sectionmark}[1]{ \markright{#1}{} }
```

Now `\leftmark` and `\rightmark` will just print the name of the chapter and section, without number and without affecting the formatting. Note that these redefinitions must be inserted *after* the first call of `\pagestyle{fancy}`. The standard book formatting of the `\chaptermark` is:

```
\renewcommand{\chaptermark}[1]{\markboth{\MakeUppercase{\chaptername\
 \thechapter.\ #1}}{}}
```

Watch out: if you provide long text in two different ”parts” only in the footer or only in the header, you might see overlapping text.

Moreover, with the following commands you can define the thickness of the decorative lines on both the header and the footer:

```
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
```

The first line for the header, the second for the footer. Setting it to zero means that there will be no line.

Plain pages issue

An issue to look out for is that the major sectioning commands (`\part`, `\chapter` or `\maketitle`) specify a `\thispagestyle{plain}`. So, if you wish to suppress all styles by inserting a `\pagestyle{empty}` at the beginning of your document, then the style command at each section will override your initial rule, for those pages only. To achieve the intended result one can follow the new section commands with `\thispagestyle{empty}`. The `\part` command, however, cannot be fixed this way, because it sets the page style, but also advances to the next page, so that `\thispagestyle{}` cannot be applied to that page. Two solutions:

- simply write `\usepackage{nopageno}` in the preamble. This package will make `\pagestyle{plain}` have the same effect as `\pagestyle{empty}`, effectively suppressing page numbering when it is used.
- Use `fancyhdr` as described below.

The tricky problem when customizing headers and footers is to get things like running section and chapter names in there. Standard 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 (`\chaptermark`, `\sectionmark`, or `\subsectionmark`) that is responsible for redefining `\rightmark` and `\leftmark`, except if they are starred -- in such a case, `\markboth{Chapter/Section name}{}` must be used inside the sectioning command if header and footer lines are to be updated.

Again, several packages provide a solution:

- an alternative one-stage mechanism is provided by the package `titleps`);
- `fancyhdr` will handle the process its own way.

16.7.2. Customizing with *fancyhdr*

To get better control over the headers, one can use the package `fancyhdr` written by Piet van Oostrum. It provides several commands that allow you to customize the header and footer lines of your document. For a more complete guide, the author of the package produced this documentation[11].

11 `http://www.ctan.org/tex-archive/macros/latex/contrib/fancyhdr/fancyhdr.pdf`

To begin, add the following lines to your preamble:

```
\usepackage{fancyhdr}
\setlength{\headheight}{15.2pt}
\pagestyle{fancy}
```

You can now observe a new style in your document.

The `\headheight` needs to be 13.6pt or more, otherwise you will get a warning and possibly formatting issues. Both the header and footer comprise three elements each according to its horizontal position (left, centre or right).

The styles supported by `fancyhdr`:

- the four LaTeX styles;
- `fancy` defines a new header for all pages but *plain-style* pages such as chapters and titlepage;
- `fancyplain` is the same, but for absolutely all pages.

Style customization

The styles can be customized with `fancyhdr` specific commands. Those two styles may be configured directly, whereas for LaTeX styles you need to make a call to the `\fancy-pagestyle` command.

To set header and footer style, `fancyhdr` provides three interfaces. They all provide the same features, you just use them differently. Choose the one you like most.

- You can use the following six commands.

```
\lhead[<even output>]{<odd output>}
\chead[<even output>]{<odd output>}
\rhead[<even output>]{<odd output>}

\lfoot[<even output>]{<odd output>}
\cfoot[<even output>]{<odd output>}
\rfoot[<even output>]{<odd output>}
```

Hopefully, the behaviour of the above commands is fairly intuitive: if it has *head* in it, it affects the head etc, and obviously, *l* , *c* and *r* means **l** eft, **c** entre and **r** ight respectively.

- You can also use the command `\fancyhead` for header and `\fancyfoot` for footer. They work in the same way, so we'll explain only the first one. The syntax is:

```
\fancyhead[selectors]{output you want}
```

You can use multiple selectors optionally separated by a comma. The selectors are the following:

`E`	even page
`O`	odd page
`L`	left side
`C`	centered
`R`	right side

so `CE,RO` will refer to the center of the even pages and to the right side of the odd pages.

- `\fancyhf` is a merge of `\fancyhead` and `\fancyfoot`, hence the name. There are two additional selectors `H` and `F` to specify the header or the footer, respectively. If you omit the `H` and the `F`, it will set the fields for both.

These commands will only work for `fancy` and `fancyplain`. To customize LaTeX default style you need the `\fancyplainstyle` command. See below for examples.

For a clean customization, we recommend you start from scratch. To do so you should *erase* the current pagestyle. Providing empty values will make the field blank. So

```
\fancyhf{}
```

will just delete the current heading/footer configuration, so you can make your own.

Plain pages

There are two ways to change the style of plain pages like chapters and titlepage.

First you can use the `fancyplain` style. If you do so, you can use the command `\fancyplain{...}{...}` inside `fancyhdr` commands like `\lhead{...}`, etc.

When LaTeX wants to create a page with an empty style, it will insert the first argument of `\fancyplain`, in all the other cases it will use the second argument. For instance:

```
\pagestyle{fancyplain}
\fancyhf{}
\lhead{ \fancyplain{}{Author Name} }
\rhead{ \fancyplain{}{\today} }
\rfoot{ \fancyplain{}{\thepage} }
```

It has the same behavior of the previous code, but you will get empty header and footer in the title and at the beginning of chapters.

Alternatively you could redefine the *plain* style, for example to have a really plain page when you want. The command to use is `\fancypagestyle{plain}{...}` and the argument can contain all the commands explained before. An example is the following:

```
\pagestyle{fancy}

\fancypagestyle{plain}{ %
  \fancyhf{} % remove everything
  \renewcommand{\headrulewidth}{0pt} % remove lines as well
  \renewcommand{\footrulewidth}{0pt}
}
```

In that case you can use any style but `fancyplain` because it would override your redefinition.

Examples

For two-sided, it's common to mirror the style of opposite pages, you tend to think in terms of *inner* and *outer* . So, the same example as above for two-sided is:

```
\lhead[Author Name]{}
\rhead[]{Author Name}
\lhead[]{\today}
\rhead[\today]{}
\lfoot[\thepage]{}
\rfoot[]{\thepage}
```

This is effectively saying author name is top outer, today's date is top inner, and current page number is bottom outer. Using `\fancyhf` can make it shorter:

```
\fancyhf[HLE,HRO]{Author's Name}
\fancyhf[HRE,HLO]{\today}
\fancyhf[FLE,FRO]{\thepage}
```

Here is the complete code of a possible style you could use for a two-sided document:

```
\usepackage{fancyhdr}
\setlength{\headheight}{15pt}

\pagestyle{fancy}
\renewcommand{\chaptermark}[1]{ \markboth{#1}{} }
\renewcommand{\sectionmark}[1]{ \markright{#1} }

\fancyhf{}
\fancyhead[LE,RO]{\thepage}
\fancyhead[RE]{\textit{ \nouppercase{\leftmark}} }
\fancyhead[LO]{\textit{ \nouppercase{\rightmark}} }

\fancypagestyle{plain}{ %
  \fancyhf{} % remove everything
  \renewcommand{\headrulewidth}{0pt} % remove lines as well
  \renewcommand{\footrulewidth}{0pt}
}
```

Using `\fancypagestyle` one can additionally define multiple styles for one's document that are easy to switch between. Here's a somewhat complicated example for a two-sided book style:

```
\fancypagestyle{fancybook}{%
    \fancyhf{}%
    % Note the ## here. It's required because \fancypagestyle is making a macro
 (\ps@fancybook).
    % If we just wrote #1, TeX would think that it's the argument to
 \ps@fancybook, but
    % \ps@fancybook doesn't take any arguments, so TeX would complain with an
 error message.
    % You are not expected to understand this.
    \renewcommand*{\sectionmark}[1]{ \markright{\thesection\ ##1} }%
    \renewcommand*{\chaptermark}[1]{ \markboth{\chaptername\ \thechapter: ##1}{}
 }%
    % Increase the length of the header such that the folios
    % (typography jargon for page numbers) move into the margin
    \fancyhfoffset[LE]{6mm}% slightly less than 0.25in
    \fancyhfoffset[RO]{6mm}%
    % Put some space and a vertical bar between the folio and the rest of the
 header
    \fancyhead[LE]{\thepage\hskip3mm\vrule\hskip3mm\leftmark}%
    \fancyhead[RO]{\rightmark\hskip3mm\vrule\hskip3mm\thepage}%
}
```

16.7.3. Page *n* of *m*

Some people like to put the current page number in context with the whole document. LaTeX only provides access to the current page number. However, you can use the `lastpage` package to find the total number of pages, like this:

```
\usepackage{lastpage}
...
\cfoot{\thepage\ of \pageref{LastPage} }
```

Note the capital letters . Also, add a backslash after `\thepage` to ensure adequate space between the page number and 'of'. And recall, when using references, that you have to run LaTeX an extra time to resolve the cross-references.

16.7.4. Alternative packages

Other packages for page styles are `scrpage2`, very similar to `fancyhdr`, and `titleps`, which takes a one-stage approach, without having to use `\leftmark` or `\rightmark`.

16.8. Page background

The `eso-pic` package will let you print content in the background of every page or individual pages.

```
\usepackage{tikz} % for \gradientbox below.
\usepackage{eso-pic}

\newcommand{\gradientbox}[3]{%
  \begin{tikzpicture}
    \node[left color=#1,right color=#2] {#3};
  \end{tikzpicture}%
}

\AddToShipoutPicture*{%
  \AtPageLowerLeft{%
    \rotatebox{90}{
      \gradientbox{blue!20}{white}{%
        \begin{minipage}{\paperheight}%
          \hspace*{ \stretch{1} }\textcopyright~2013
 \makeatletter\@author\makeatother.\hspace*{ \stretch{1} }
        \end{minipage}%
      }
    }%
  }%
}
```

The starred-version of the `\AddToShipoutPicture` command applies to the current page only.

16.9. Multi-column pages

16.9.1. Using the *twocolumn* optional class argument

Using a standard Latex document class, like article, you can simply pass the optional argument *twocolumn* to the document class: `\documentclass[twocolumn]{article}` which will give the desired effect.

While this approach is useful, it has limitations:

- Can support up to ten columns.
- Implements a *multicols* environment, therefore, it is possible to mix the number of columns within a document.
- Additionally, the environment can be nested inside other environments, such as `figure`.
- `multicol` outputs *balanced* columns, whereby the columns on the final page will be of roughly equal length.
- Vertical rules between columns can be customised.
- Column environments can be easily customised locally or globally.

16.9.2. Using `multicol` package

The `multicol` package overcomes some of the shortcomings of *twocolumn* and provides the `multicol` environment. To create a typical two-column layout:

```
\begin{multicols}{2}
  lots of text
\end{multicols}
```

Floats are not fully supported by this environment. It can only cope if you use the starred forms of the float commands (e.g., `\begin{figure*}`) which makes the float span all columns. This is not hugely problematic, since floats of the same width as a column may be too small, and you would probably want to span them anyway. See this section[12] for a more detailed discussion.

The `multicol` package has two important parameters which can be set[13] using `\setlength`:

- `\columnseprule`, sets the width of the vertical rule between columns and defaults to 0pt
- `\columnsep`, sets the horizontal space between columns and the defaults to 10pt, which is quite narrow

To force a break in a column, the command `\columnbreak` is used.

16.10. Manual page formatting

There may be instances, especially in very long documents, such as books, that LaTeX will not get all page breaks looking as good as it could. It may, therefore, be necessary

12 Chapter 18.8 on page 244
13 Chapter 23.3 on page 284

to manually tweak the page formatting. Of course, you should only do this at the very final stage of producing your document, once all the content is complete. LaTeX offers the following:

`\newpage`	Ends the current page and starts a new one.
`\pagebreak[number]`	Breaks the current page at the point of the command. The optional *number* argument sets the priority in a scale from 0 to 4.
`\nopagebreak[number]`	Stops the page being broken at the point of the command. The optional *number* argument sets the priority in a scale from 0 to 4.
`\clearpage`	Ends the current page and causes any floats encountered in the input, but yet to appear, to be printed.

16.11. Widows and orphans

w:Widows and orphans[14] In professional books, it's not desirable to have single lines at the beginning or end of a page. In typesetting such situations are called 'widows' and 'orphans'. Normally it is possible that widows and orphans appear in LaTeX documents. You can try to deal with them using manual page formatting, but there's also an automatic solution.

LaTeX has a parameter for 'penalty' for widows and orphans ('club lines' in LaTeX terminology). With the greater penalty LaTeX will try more to avoid widows and orphans. You can try to increase these penalties by putting following commands in your document preamble:

```
\widowpenalty=300
\clubpenalty=300
```

If this does not help, you can try increasing these values even more, to a maximum of 10000. However, it is not recommended to set this value too high, as setting it to 10000 forbids LaTeX from doing this altogether, which might result in strange behavior.

It also helps to have rubber band values for the space between paragraphs:

```
\setlength{\parskip}{3ex plus 2ex minus 2ex}
```

Alternatively, you can use the `needspace` package to *reserve* some lines and thus to prevent page breaking for those lines.

```
\needspace{5\baselineskip}
Some
text
on
5
lines.
```

14 `http://en.wikipedia.org/wiki/Widows%20and%20orphans`

16.12. Troubleshooting

A very useful troubleshooting and designing technique is to turn on the showframe option in the geometry package (which has the same effect as the showframe package described above. It draws bounding boxes around the major page elements, which makes where the various regions of the page are, which is often unclear because of white whitespace.

```
\usepackage[showframe]{geometry}
```

16.13. Notes and References

This page uses material from Andy Roberts' Getting to grips with LaTeX with permission from the author.

17. Importing Graphics

There are two possibilities to include graphics in your document. Either create them with some special code, a topic which will be discussed in the *Creating Graphics* part, (see Introducing Procedural Graphics[1]) or import productions from third party tools[2], which is what we will be discussing here.

Strictly speaking, LaTeX cannot manage pictures directly: in order to introduce graphics within documents, LaTeX just creates a box with the same size as the image you want to include and embeds the picture, without any other processing. This means you will have to take care that the images you want to include are in the right format to be included. This is not such a hard task because LaTeX supports the most common picture formats around.

17.1. Raster graphics vs. vector graphics

Raster graphics will highly contrast with the quality of the document if they are not in a high resolution, which is the case with most graphics. The result may be even worse once printed.

Most drawing tools (e.g. for diagrams) can export in vector format. So you should always prefer PDF or EPS to PNG or JPG.

17.2. The *graphicx* package

As stated before, LaTeX can't manage pictures directly, so we will need some extra help: we have to load the `graphicx` package`http://ctan.org/pkg/graphicx/` in the preamble of our document:

```
\usepackage{graphicx}
```

This package accepts as an argument the external driver to be used to manage pictures; however, the latest version of this package takes care of everything by itself, changing the driver according to the compiler you are using, so you don't have to worry about this. Still, just in case you want to understand better how it works, here are the possible options you can pass to the package:

- `dvips` (default if compiling with `latex`), if you are compiling with `latex` to get a DVI and you want to see your document with a DVI or PS viewer.

1 Chapter 44 on page 517
2 Chapter 17.12 on page 226

- `dvipdfm`, if you are compiling with `latex` to get a DVI that you want to convert to PDF using *dvipdfm* , to see your document with any PDF viewer.
- `pdftex` (default if compiling with `pdflatex`), if you are compiling with `pdftex` to get a PDF that you will see with any PDF viewer.

But, again, you don't need to pass any option to the package because the default settings are fine in most of the cases.

In many respects, importing your images into your document using LaTeX is fairly simple... *once* you have your images in the right format that is! Therefore, I fear for many people the biggest effort will be the process of converting their graphics files. Now we will see which formats we can include and then we will see how to do it.

17.3. Document Options

The graphics and graphicx packages recognize the `draft` and `final` options given in the `\documentclass[...]{...}` command at the start of the file. (See Document Classes[3].) Using `draft` as the option will suppress the inclusion of the image in the output file and will replace the contents with the name of the image file that would have been seen. Using `final` will result in the image being placed in the output file. The default is `final`.

17.4. Supported image formats

As explained before, the image formats you can use depend on the driver that `graphicx` is using but, since the driver is automatically chosen according to the compiler, then the allowed image formats will depend on the compiler you are using.

⚠ Warning

Using `pdflatex` will be usually much more simple for graphics inclusion as it supports widespread formats such as PDF, PNG and JPG. Read this chapter carefully if you are using the DVI compiler (`latex`), otherwise you might encounter a lot of errors at compile time.

Consider the following situation: you have added some pictures to your document in JPG and you have successfully compiled it in PDF. Now you want to compile it in DVI, you run `latex` and you get a lot of errors... because you forgot to provide the EPS versions of the pictures you want to insert.

At the beginning of this book, we had stated that the same LaTeX source can be compiled in both DVI and PDF without any change. This is true, as long as you don't use particular packages, and `graphicx` is one of those. In any case, you can still use both compilers with documents with pictures as well, as long as you always remember to provide the pictures in two formats (EPS and one of JPG, PNG or PDF).

3 Chapter 5.2.1 on page 52

17.4.1. Compiling with *latex*

The only format you can include while compiling with `latex` is Encapsulated PostScript[4] (**EPS**).

The EPS format was defined by Adobe Systems for making it easy for applications to import postscript-based graphics into documents. Because an EPS file declares the size of the image, it makes it easy for systems like LaTeX to arrange the text and the graphics in the best way. EPS is a vector format[5]—this means that it can have very high quality if it is created properly, with programs that are able to manage vector graphics. It is also possible to store bit-map pictures within EPS, but they will need *a lot* of disk space.

17.4.2. Compiling with *pdflatex*

If you are compiling with `pdflatex` to produce a PDF, you have a wider choice. You can insert:

- **JPG** , widely used on Internet, digital cameras, etc. They are the best choice if you want to insert photos.
- **PNG** , a very common format (even if not as much as JPG); it's a lossless[6] format and it's the best choice for diagrams (if you were not able to generate a vector[7] version) and screenshots.
- **PDF** , it is widely used for documents but can be used to store images as well. It supports both vector and bit-map[8] images, but it's not recommended for the latter, as JPG or PNG will provide the same result using less disk space.
- **EPS** can be used with the help of the `epstopdf` package. Depending on your installation,
  - you may just need to have it installed, there is no need to load it in your document;
  - if it does not work, you need to load it just after the `graphicx` package. Additionally, since `epstopdf` will need to convert the EPS file into a PDF file and store it, you need to give "writing permissions" to your compiler. This is done by adding an option to the compiling command, *e.g.* `pdflatex -shell-escape file.tex` (if you use a LaTeX editor, they usually allow to modify the command in the configuration options). Check the epstopdf documentation for other compilers.

17.5. Including graphics

Now that we have seen which formats we can include and how we could manage those formats, it's time to learn how to include them in our document. After you have loaded the `graphicx` package in your preamble, you can include images with `\includegraphics`, whose syntax is the following:

4 `http://en.wikipedia.org/wiki/Encapsulated%20PostScript`
5 `http://en.wikipedia.org/wiki/Vector%20graphics`
6 `http://en.wikipedia.org/wiki/lossless`
7 `http://en.wikipedia.org/wiki/Vector%20graphics`
8 `http://en.wikipedia.org/wiki/Raster%20graphics`

```
\includegraphics[attr1=val1, attr2=val2, ..., attrn=valn]{imagename}
```

As usual, arguments in square brackets are optional, whereas arguments in curly braces are compulsory.

The argument in the curly braces is the name of the image. Write it *without* the extension. This way the LaTeX compiler will look for any supported image format in that directory and will take the best one (EPS if the output is DVI; JPEG, PNG or PDF if the output is PDF). Images can be saved in multiple formats for different purposes. For example, a directory can have "`diagram.pdf` " for high-resolution printing, while "`diagram.png` " can be used for previewing on the monitor. You can specify which image file is to be used by `pdflatex` through the preamble command:

```
\DeclareGraphicsExtensions{.pdf,.png,.jpg}
```

which specifies the files to include in the document (in order of preference), if files with the same basename exist, but with different extensions.

The variety of possible attributes that can be set is fairly large, so only the most common are covered below:

`width=xx`	Specify the preferred width of the imported image to *xx* .	*NB. Only specifying either width or height will scale the image while maintaining the aspect ratio.*
`height=xx`	Specify the preferred height of the imported image to *xx* .	
`keepaspectratio`	This can be set to either *true* or *false* . When true, it will scale the image according to both height and width, but will not distort the image, so that neither width nor height are exceeded.	
`scale=xx`	Scales the image by the desired scale factor. e.g, 0.5 to reduce by half, or 2 to double.	
`angle=xx`	This option can rotate the image by *xx* degrees (counter-clockwise)	
`trim=l b r t`	This option will crop the imported image by *l* from the left, *b* from the bottom, *r* from the right, and *t* from the top. Where l, b, r and t are lengths.	
`clip`	For the `trim` option to work, you must set `clip=true`.	
`page=x`	If the image file is a pdf file with multiple pages, this parameter allows you to use a different page than the first.	
`resolution=x`	Specify image resolution in dpi	

In order to use more than one option at a time, simply separate each with a comma. The order you give the options matters. E.g you should first rotate your graphic (with angle) and then specify its width.

Included graphics will be inserted just *there* , where you placed the code, and the compiler will handle them as "big boxes". As we will see in the floats section[9], this can disrupt the layout; you'll probably want to place graphics inside floating objects.

9 Chapter 18 on page 231

Also note that the trim option does not work with XeLaTex.

Be careful using any options, if you are working with the chemnum-package. The labels defined by `\cmpdref{<label name>}` might not behave as expected. Scaling the image for instance may be done by `\scalebox` instead.

The *star* version of the command will work for .eps files only. For a more portable solution, the standard way should take precedence. The star command will take the crop dimension as extra parameter:

```
\includegraphics*[100,100][300,300]{mypicture}
```

17.5.1. Examples

OK, it's time to see graphicx in action. Here are some examples. Say you had a file 'chick.jpg' you would include it like:

```
\includegraphics{chick}
```

This simply imports the image, without any other processing. However, it is very large (so we won't give a example of how it would look here!) So, let's scale it down:

```
 \includegraphics[scale=0.5]{chick}
```

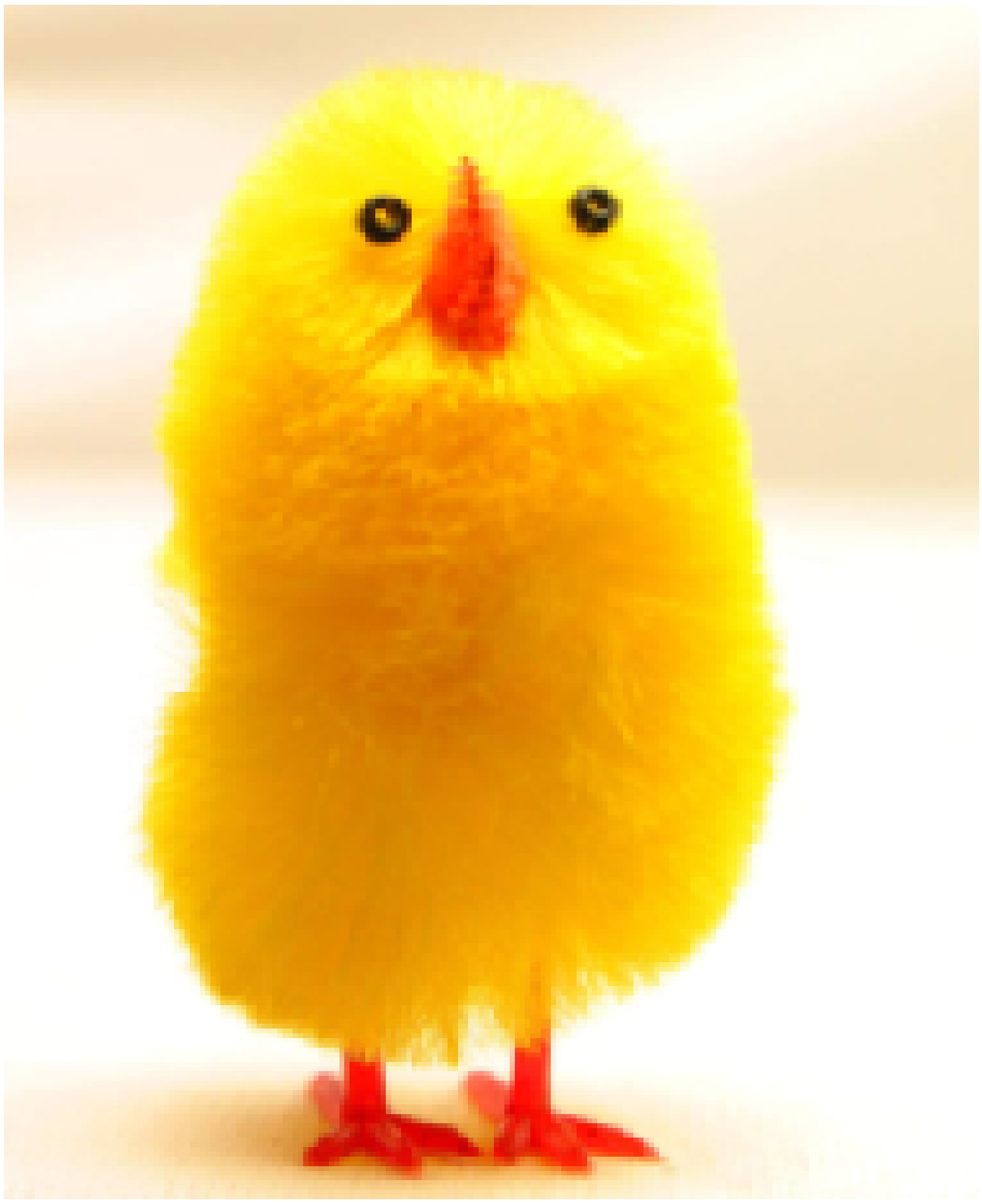


Figure 62

This has now scaled it by half. If you wish to be more specific and give actual lengths of the image dimensions, this is how to go about it:

```
\includegraphics[width=2.5cm]{chick}
```

One can also specify the scale with respect to the width of a line in the local environment (`\linewidth`), the width of the text on a page (`\textwidth`) or the height of the text on a page (`\textheight`) (pictures not shown):

```
\includegraphics[width=\linewidth]{chick}
\includegraphics[width=\textwidth]{chick}
\includegraphics[height=\textheight]{chick}
```

To rotate (I also scaled the image down):

```
\includegraphics[scale=0.5, angle=180]{chick}
```

And finally, an example of how to crop an image should you wish to focus on one particular area of interest:

```
%trim option's parameter order: left bottom right top
\includegraphics[trim = 10mm 80mm 20mm 5mm, clip, width=3cm]{chick}
```

Note: the presence of `clip`, as the trim operation will not work without it.

Trick: You can also use negative trim values to add blank space to your graphics, in cases where you need some manual alignment.

17.5.2. Spaces in names

If the image file were called ”chick picture.png”, then you need to include the full filename when importing the image:

```
\includegraphics[scale=0.5]{chick picture.png}
```

chick picture.png

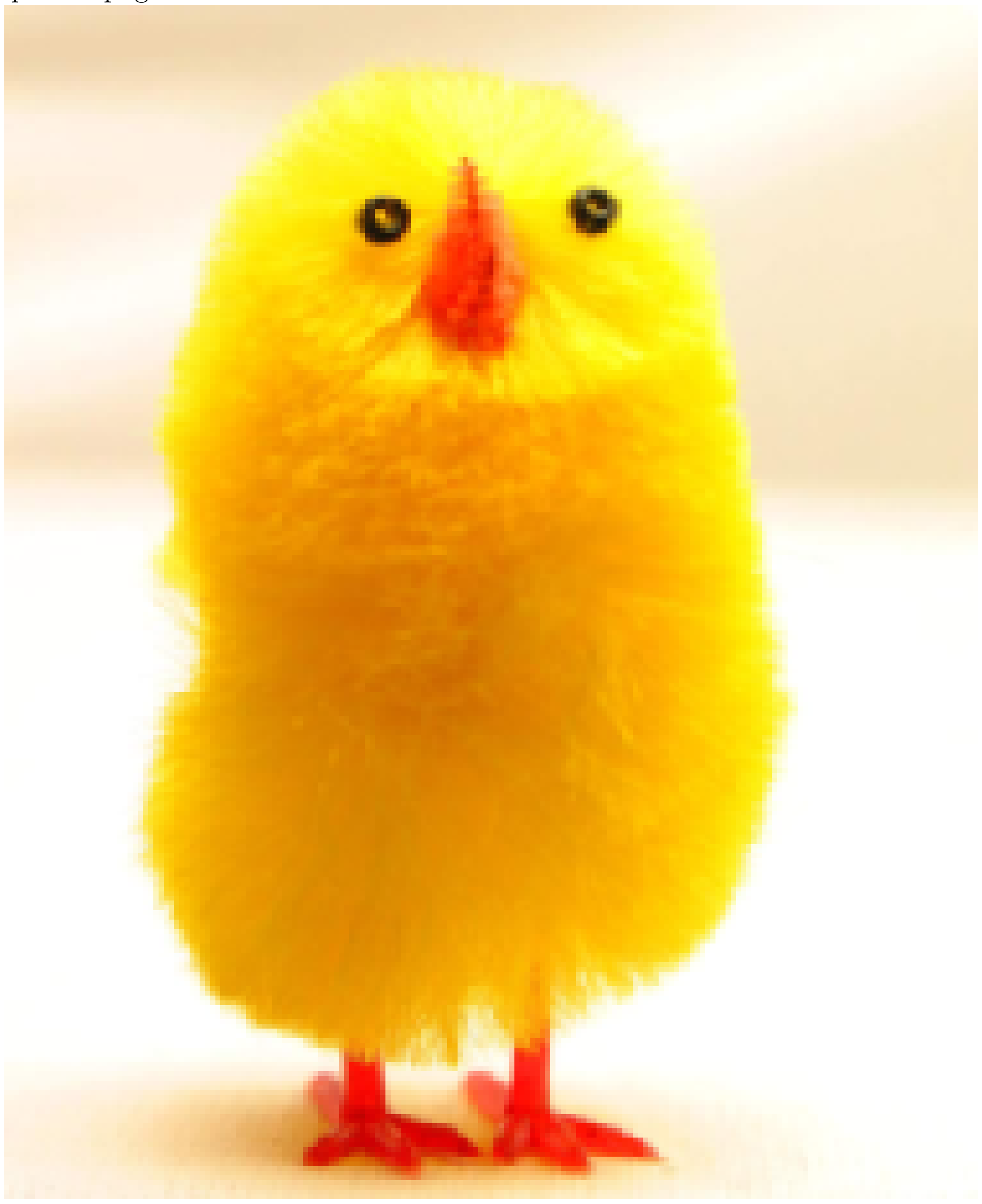


Figure 63

One option is to not use spaces in file names (if possible), or to simply replace spaces with underscores (”chick picture.png” to ”chick_picture.png”)

```
\includegraphics[scale=0.5]{chick_picture.png}
```

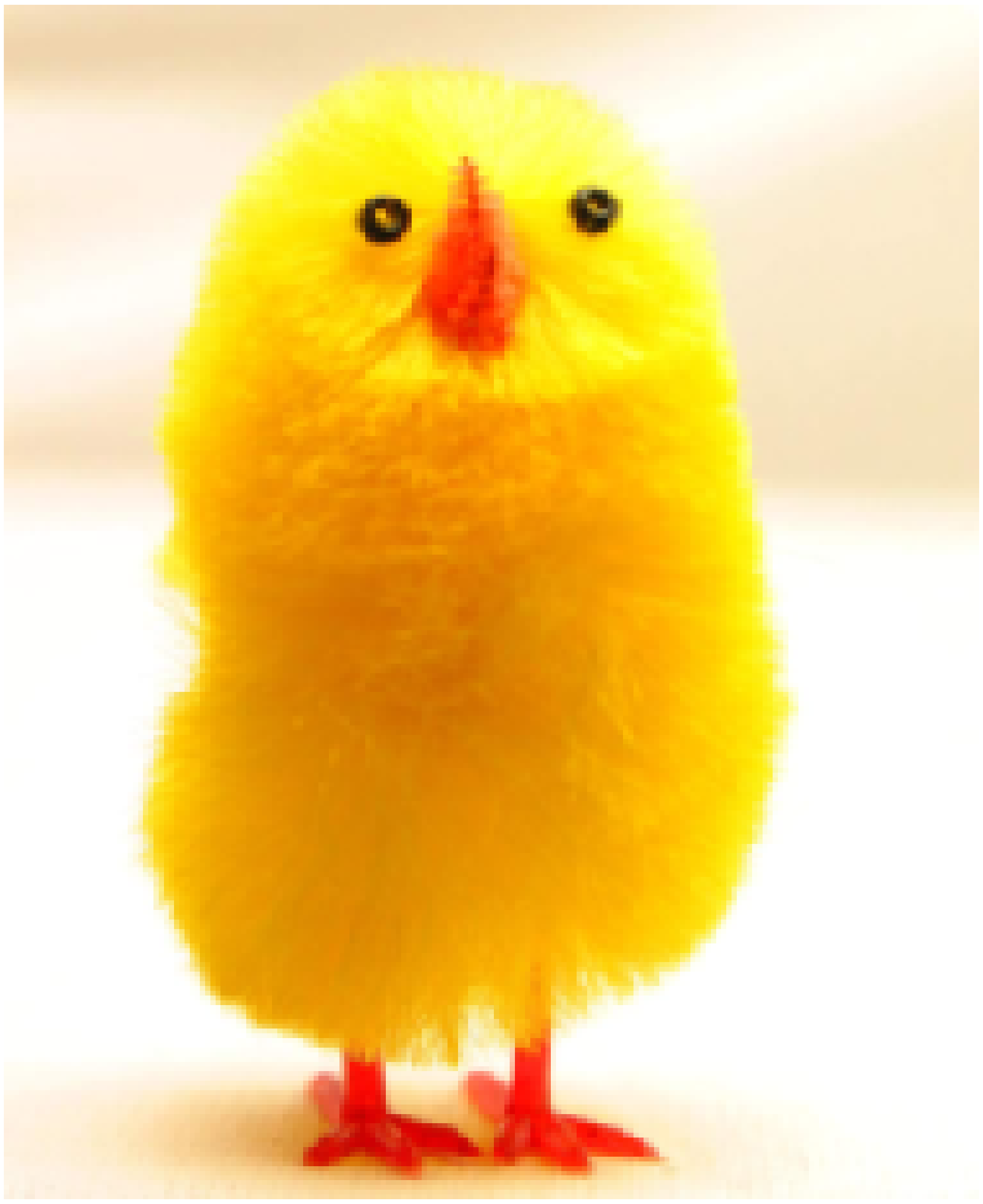


Figure 64

17.5.3. Borders

It is possible to have LaTeX create a border around your image by using `\fbox`:

```
\setlength\fboxsep{0pt}
\setlength\fboxrule{0.5pt}
\fbox{\includegraphics{chick}}
```

You can control the border padding with the `\setlength\fboxsep{0pt}` command, in this case I set it to 0pt to avoid any padding, so the border will be placed tightly around the image. You can control the thickness of the border by adjusting the `\setlength\fboxrule{0.5pt}` command.

See Boxes[10] for more details on `\framebox` and `\fbox`.

17.6. Graphics storage

The command `\graphicspath` tells LaTeX where to look for images, which can be useful if you store images centrally for use in many different documents. The `\graphicspath` command takes one argument, which specifies the additional paths you want to be searched when the `\includegraphics` command is used. Here are some examples (trailing / is required):

```
\graphicspath{ {/var/lib/images/} }
\graphicspath{ {images_folder/}{other_folder/}{third_folder/} }
\graphicspath{ {./images/} }
\graphicspath{ {c:\mypict~1\camera} }
\graphicspath{ {c:/mypict~1/camera/} } % works well in Win XP
```

Please see `http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.pdf`. In the third example shown there should be a directory named ”images” in the same directory as your main tex file, i.e. this is RELATIVE addressing.

Using absolute paths, `\graphicspath` makes your file less portable, while using relative paths (like the third example), there should not be any problem with portability. The fourth example uses the ”safe” (MS-DOS) form of the Windows *MyPictures* folder because it’s a bad idea to use directory names containing spaces. Again, ensure file names do not contain spaces or alternatively if you are using PDFLaTeX, you can use the package `grffile` which will allow you to use spaces in file names.

Note that you cannot make the graphicx package search directories recursively. Under Linux/Unix, you can achieve a recursive search using the environment variable `TEXINPUTS` , e.g., by setting it to

```
export TEXINPUTS=./images//:./Snapshots//
```

before running latex/pdflatex or your TeX-IDE. (But this, of course, is not a portable method.)

17.7. Images as figures

The figure environment is not exclusively used for images. We will only give a short preview of figures here. More information on the figure environment and how to use it can be found in Floats, Figures and Captions[11].

10 Chapter 25 on page 295
11 Chapter 18 on page 231

There are many scenarios where you might want to accompany an image with a caption and possibly a cross-reference. This is done using the `figure` environment. The following code sample shows the bare minimum required to use an image as a figure.

```
\begin{figure}[p]
    \includegraphics{image.png}
\end{figure}
```

The above code extract is relatively trivial, and doesn't offer much functionality. The following code sample shows an extended use of the figure environment which is almost universally useful, offering a caption and label, centering the image and scaling it to 80% of the width of the text.

```
\begin{figure}[p]
    \centering
    \includegraphics[width=0.8\textwidth]{image.png}
    \caption{Awesome Image}
    \label{fig:awesome_image}
\end{figure}
```

17.8. Text wrapping around pictures

See Floats, Figures and Captions[12].

17.9. Seamless text integration

The drawback of importing graphics that were generated with a third-party tool is that font and size will not match with the rest of the document. There are still some workarounds though.

The easiest solution is to use the picture environment and then simply use the ”put” command to put a graphics file inside the picture, along with any other desired LaTeX element. For example:

```
 \setlength{\unitlength}{0.8cm}
 \begin{picture}(6,5)
 \put(3.5,0.4){$\displaystyle
 s:=\frac{a+b+c}{2}$}
 \put(1,1){\includegraphics[
   width=2cm,height=2cm]{picture.eps} }
 \end{picture}
```

12 Chapter 18 on page 231

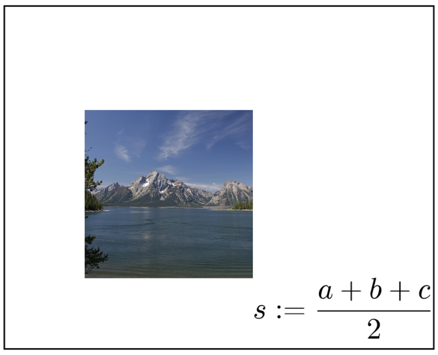


Figure 65

Note that the border around the picture in the above example was added by using `\fbox` [13], so the contents of the border is the picture as generated by the above code.

Tools like Inkscape or Xfig have a dedicated LaTeX export feature that will let you use correct font and size for text in vector graphics. See #Third-party graphics tools[14].

For a perfect integration of graphics, you might consider procedural graphics[15] capabilities of some LaTeX packages like TikZ or PSTricks. It lets you *draw* from within a document source. While the learning curve is steeper, it is worth it most of the time.

17.10. Including full PDF pages

There is a great package for including full pages of PDF files: pdfpages[16]. It is capable of inserting entire pages as is and more pages per one page in any layout (e.g. 2x3).

The package has several options:

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

Options:

- **final** : Inserts pages. This is the default.
- **draft** : Does not insert pages, but prints a box and the filename instead.
- **enable-survey** : Activates survey functionalities. (Experimental, subject to change.)

The first command is

```
\includepdf[ key=val ]{ filename }
```

Options for **key=val** (A comma separated list of options using the key = value syntax)

13 Chapter 17.5.3 on page 219
14 Chapter 17.12 on page 226
15 Chapter 44 on page 517
16 `http://www.ctan.org/tex-archive/macros/latex/contrib/pdfpages`

pages	Selects pages to insert. The argument is a comma separated list, containing page numbers (pages={3,5,6,8}), ranges of page numbers (pages={4-9}) or any combination. To insert empty pages use {}. For instance `pages={3,{},8-11,15}` will insert page 3, an empty page, and pages 8, 9, 10, 11, and 15.Actually not only links but all kinds of PDF annotations will get lost. Page ranges are specified by the following syntax: *m* - *n* . This selects all pages from *m to n* . Omitting m defaults to the first page; omitting n defaults to the last page of the document. Another way to select the last page of the document, is to use the keyword last. (This is only permitted in a page range.)E.g.: *pages=-* will insert all pages of the document, and *pages=last-1* will insert all pages in reverse order.(Default: pages=1)
angle	You can use the angle-option for turning the included page, for exampe for turning a landscape document when the latex-document is portrait. Example: `angle=90`
addtolist	Adds an entry to the list of figures, the list of tables, or any other list (e.g. from float.sty). This option requires four arguments, separated by commas:`addtolist={ page number , type , heading , label }` • `page number` : Page number of the inserted page. • `type`: Name of a floating environment. (figure, table, etc.) • `heading`: Title inserted into LoF, LoT, etc. • `label`: Name of the label. This label can be referred to with \ref and \pageref. Like addtotoc, addtolist accepts multiple sets of the above mentioned four arguments, all separated by commas. The proper recursive definition is:`addtolist={ page number , type , heading , label [, lof-list ] }`
pagecommand	Declares LaTeX-commands, which are executed on each sheet of paper. (Default: *pagecommand={\thispagestyle{empty}}* `pagecommand={\label{fig:mylabel}}`

You can also inserts pages of several external PDF documents.

```
\includepdfmerge[ key=val ]{ file-page-list }
```

Several PDFs can be placed table-like on one page. See more information in its documentation[17].

17.11. Converting graphics

Note

17 `http://www.ctan.org/tex-archive/macros/latex/contrib/pdfpages/pdfpages.pdf`

You should also take a look at Export To Other Formats[18] for other possibilities.

epstopdf

You can convert EPS to PDF with the epstopdf utility[19], included in package of the same name. This tool is actually called by `pdflatex` to convert EPS files to PDF in the background when the `graphicx` package is loaded. This process is completely invisible to the user.

You can batch convert files using the command-line. In Bourne Shell (Unix) this can be done by:

```
$ for i in *.eps; do epstopdf "$i"; done
```

In Windows, multiple files can be converted by placing the following line in a batch file[20] (a text file with a `.bat` extension) in the same directory as the images:

```
for %%f in (*.eps) do epstopdf %%f
```

which can then be run from the command line.

If `epstopdf` produces whole page with your small graphics somewhere on it, use

```
$ epstopdf --gsopt=-dEPSCrop foo.eps
```

or try using `ps2pdf` utility which should be installed with Ghostscript (required for any TeX distribution).

```
$ ps2pdf -dEPSCrop foo.eps
```

to crop final PDF.

eps2eps

When all of the above fails, one can simplify the EPS file before attempting other conversions, by using the eps2eps[21] tool (also see next section):

```
$ eps2eps input.eps input-e2.eps
```

This will convert all the fonts to pre-drawn images, which is sometimes desirable when submitting manuscripts for publication. However, on the downside, the fonts are NOT converted to lines, but instead to bitmaps, which reduces the quality of the fonts.

imgtops

imgtops[22] is a lightweight graphics utility for conversions between raster graphics (JPG, PNG, ...) and EPS/PS files.

Inkscape

18 Chapter 57 on page 627
19 `http://www.ctan.org/tex-archive/support/epstopdf/`
20 `http://en.wikipedia.org/wiki/Batch%20file`
21 `http://linuxcommand.org/man_pages/eps2eps1.html`
22 `http://imgtops.sourceforge.net/`

Inkscape can also convert files from and to several formats, either from the GUI or from the command-line. For instance, to obtain a PDF from a SVG image you can do:

```
$ inkscape -z -D --file=input.svg --export-pdf=output.pdf
```

It is possible to run this from within a LaTeX file, the

UNKNOWN TEMPLATE LaTeX/package

svg

package (when running (pdf)latex with the --shell-escape option) can do this using Inkscape's pdf+tex export option, or a simple macro can be used. See How to include SVG diagrams in LaTeX? -- Stackexchange[23] See Export To Other Formats[24] for more details.

pstoedit

To properly edit an EPS file, you can convert it to an *editable* format using pstoedit[25]. For instance, to get an Xfig-editable file, do:

```
$ pstoedit -f fig input.eps output.fig
```

And to get an SVG file (editable with any vector graphics tool like Inkscape) you can do:

```
$ pstoedit -f plot-svg input.eps output.svg
```

Sometimes pstoedit fails to create the target format (for example when the EPS file contains clipping information).

PDFCreator

Under Windows, PDFCreator[26] is an open source software that can create PDF as well as EPS files. It installs a virtual printer that can be accessed from other software having a "print..." entry in their menu (virtually any program).

Raster graphics converters

- Sam2p[27] (`convert`) or
- ImageMagick[28] (`convert`) or
- GraphicsMagick[29] (`gm convert`).

These three programs operate much the same way, and can convert between most graphics formats. Sam2p however is the most recent of the three and seems to offer both the best quality and to result in the smallest files.

23 `http://tex.stackexchange.com/q/2099/28808`
24 Chapter 57 on page 627
25 `http://www.pstoedit.net/`
26 `http://sourceforge.net/projects/pdfcreator/`
27 `http://pts.szit.bme.hu/sam2p/`
28 `http://www.imagemagick.org/`
29 `http://www.graphicsmagick.org/`

17.11.1. PNG alpha channel

Acrobat Reader sometimes has problems with displaying colors correctly if you include graphics in PNG format with alpha channel. You can solve this problem by dropping the alpha channel. On Linux it can be achieved with `convert` from the ImageMagick[30] program:

```
convert -alpha off input.png output.png
```

17.11.2. Converting a color EPS to grayscale

Sometimes color EPS figures need to be converted to black-and-white or grayscale to meet publication requirements. This can be achieved with the eps2eps[31] of the Ghostscript[32] package and `http://www.pa.op.dlr.de/~PatrickJoeckel/pscol/index.html` programs:

```
$ eps2eps input.eps input-e2.eps
$ pscol -0gray input-e2.eps input-gray.eps
```

17.12. Third-party graphics tools

We will not tackle the topic of procedural graphics created from within LaTeX code here (TikZ, PSTricks, MetaPost and friends). See Introducing Procedural Graphics[33] for that.

You should prefer vector graphics over raster graphics for their quality. Raster graphics should only be used in case of photos. Diagrams of any sort should be vectors.

As we have seen before, LaTeX handles

- EPS and PDF for vector graphics;
- PNG and JPG for raster graphics.

If some tools cannot save in those formats, you may want to convert[34] them before importing them.

17.12.1. Vector graphics

Dia

Dia[35] is a cross platform diagramming utility which can export eps images, or generate tex drawn using the `tikz` package.

Inkscape

30 `http://en.wikipedia.org/wiki/ImageMagick`
31 `http://linuxcommand.org/man_pages/eps2eps1.html`
32 `http://ghostscript.com/`
33 Chapter 44 on page 517
34 Chapter 17.11 on page 223
35 `http://live.gnome.org/Dia`

Another program for creating vector graphics is Inkscape[36]. It can run natively under Windows, Linux or Mac OS X (with X11). It works with Scalable Vector Graphics (SVG)[37] files, although it can export to many formats that can be included in LaTeX[38] files, such as EPS and PDF. From version 0.48, there is a combined PDF/EPS/PS+LaTeX output option, similar to that offered by Xfig[39]. There are instructions[40] on how to save your vector images in a PDF format understood by LaTeX and have LaTeX manage the text styles and sizes in the image automatically.[41]. Today there is the `svg-package`[43] which provides an `\includesvg` command to convert and include svg-graphics directly in your LaTeX document using Inkscape. You may have a look at this extended example[45] too.

An extremely useful plug-in is textext[46], which can import LaTeX objects. This can be used for inserting mathematical notation or LaTeX fonts into graphics (which may then be imported into LaTeX documents).

Ipe

The Ipe[47] extensible drawing editor is a free vector graphics editor for creating figures in PDF or EPS format. Unlike Xfig, Ipe represents LaTeX[48] fonts in their correct size on the screen which makes it easier to place text labels at the right spot. Ipe also has various snapping modes (for example, snapping to points, lines, or intersections) that can be used for geometric constructions.

lpic

Yet another solution is provided by the `lpic` packages `http://www.math.uni-leipzig.de/~matveyev/lpic/`, which allows TeX annotations to imported graphics. See Labels in the figures[49].

OpenOffice.org

It is also possible to export vector graphics to EPS format using OpenOffice.org[50] Draw, which is an open source office suite available for Windows, Linux and Mac.

TpX

Vector editor TpX[51] separates geometric objects from text objects. Geometric objects are saved into .PDF file, the rest is saved in .TpX file to be processed by LaTeX. User just create the graphics in TpX editor and calls the .TpX file from latex file by command \input{...TpX}.

36 `http://www.inkscape.org/`
37 `http://www.w3.org/Graphics/SVG/`
38 `http://en.wikibooks.org/wiki/LaTeX`
39 `http://en.wikipedia.org/wiki/Xfig`
40 `http://mirrors.ctan.org/info/svg-inkscape/InkscapePDFLaTeX.pdf`
41 How to include an SVG image in LATEX[42]. mirrorcatalogs.com. Retrieved
43 The svg-package on CTAN[44]. ctan.org. Retrieved
45 `http://laclaro.wordpress.com/2013/09/09/updated-includesvg-example/`
46 `http://pav.iki.fi/software/textext/`
47 `http://en.wikipedia.org/wiki/Ipe%20%28program%29`
48 `http://en.wikibooks.org/wiki/LaTeX`
49 Chapter 18.10 on page 247
50 `http://en.wikipedia.org/wiki/OpenOffice.org`
51 `http://tpx.sourceforge.net/`

Xfig

Xfig[52] is a basic program that can produce vector graphics, which can be exported to LaTeX. It can be installed on Unix platforms.

On Microsoft Windows systems, Xfig can only be installed using Cygwin-X[53]; however, this will require a fast internet connection and about 2 gigabytes of space on your computer. With Cygwin, to run Xfig, you need to first start the ”Start X - Server”, then launch ”xterm” to bring up a terminal. In this terminal type ”xfig” (without the quotation marks) and press return.

Alternatively, WinFIG[54] is an attempt to achieve the functionality of xfig on Windows computers.

There are many ways to use xfig to create graphics for LaTeX documents. One method is to export the drawing as a LaTeX document. This method, however, suffers from various drawbacks: lines can be drawn only at angles that are multiples of 30 and 45 degrees, lines with arrows can only be drawn at angles that are multiples of 45 degrees, several curves are not supported, etc.

Exporting a file as PDF/LaTeX or PS/LaTeX, on the other hand, offers a good deal more flexibility in drawing. Here’s how it’s done:

1. Create the drawing in xfig. Wherever you need LaTeX text, such as a mathematical formula, enter a LaTeX string in a textbox.
2. Use the Edit tool to open the properties of each of those textboxes, and change the option on the ”Special Flag” field to Special. This tells LaTeX to interpret these textboxes when it opens the figure.
3. Go to File -> Export and export the file as PDF/LaTeX (both parts) or PS/LaTeX (both parts), depending on whether you are using pdflatex or pslatex to compile your file.
4. In your LaTeX document, where the picture should be, use the following, where ”test” is replaced by the name of the image:

   ```
   \begin{figure}[htbp]
    \centering
    \input{test.pdf_t}
    \caption{Your figure}
    \label{figure:example}
   \end{figure}
   ```

   Observe that this is just like including a picture, except that rather than using `\includegraphics` , we use `\input` . If the export was into PS/LaTeX, the file extension to include would be .pstex_t instead of .pdf_t.
5. Make sure to include packages `graphicx` and `color` in the file, with the `\usepackage` command right below the `\documentclass` command, like this:

   ```
   \usepackage{graphicx}
   \usepackage{color}
   ```

52 http://en.wikipedia.org/wiki/Xfig
53 http://www.cygwin.com/
54 http://www.schmidt-web-berlin.de/winfig/index.shtml

And you're done!

For more details on using xfig with LaTeX, this chapter[55] of the xfig User Manual[56] may prove helpful.

Other tools

Commercial vector graphics software, such as Adobe Illustrator, CorelDRAW, and FreeHand are commonly used and can *read* and *write* EPS figures. However, these products are limited to Windows and Mac OS X platforms.

17.12.2. Raster graphics

Adobe Photoshop

It can save to EPS.

GIMP

GIMP[57], has a graphical user interface, and it is multi-platform. It can save to EPS and PDF.

17.12.3. Plots and Charts

Generic Mapping Tools (GMT)

Generic Mapping Tools (GMT)[58], maps and a wide range of highly customisable plots.

Gnumeric

Gnumeric[59], spreadsheets has SVG, EPS, PDF export

Gnuplot

Gnuplot[60], producing scientific graphics since 1986. If you want to make mathematical plots, then Gnuplot[61] can save in any format. You can get best results when used along PGF/TikZ[62].

matplotlib

55 `http://www-epb.lbl.gov/xfig/latex_and_xfig.html`
56 `http://www-epb.lbl.gov/xfig/contents.html`
57 `http://www.gimp.org`
58 `http://gmt.soest.hawaii.edu/`
59 `http://projects.gnome.org/gnumeric/`
60 `http://www.gnuplot.info`
61 `http://www.gnuplot.info`
62 Chapter 47 on page 535

matplotlib[63], plotting library written in python, with PDF and EPS export. On the other hand there is a PGF export also. There are some tricks to be able to import formats other than EPS into your DVI document, but they're very complicated. On the other hand, converting any image to EPS is very simple, so it's not worth considering them.

R

R[64], statistical and scientific figures.

17.12.4. Editing EPS graphics

As described above, graphics content can be imported into LaTeX[65] from outside programs as EPS files. But sometimes you want to edit or retouch these graphics files. An EPS file can be edited with any text editor since it is formatted as ASCII. In a text editor, you can achieve simple operations like replacing strings or moving items slightly, but anything further becomes cumbersome. Vector graphics editors, like Inkscape, may also be able to import EPS files for subsequent editing. This approach also for easier editing. However, the importing process may occassionally modify the original EPS image.

17.13. Notes and References

63 `http://matplotlib.sourceforge.net/`
64 `http://www.r-project.org/`
65 `http://en.wikibooks.org/wiki/LaTeX`

18. Floats, Figures and Captions

The previous chapter[1] introduced importing graphics. However, just having a picture stuck in between paragraphs does not look professional. To start with, we want a way of adding captions, and to be able to cross-reference. What we need is a way of defining *figures* . It would also be good if LaTeX could apply principles similar to when it arranges text to look its best to arranging pictures as well. This is where *floats* come into play.

18.1. Floats

Floats are containers for things in a document that cannot be broken over a page. LaTeX by default recognizes ”table” and ”figure” floats, but you can define new ones of your own (see Custom floats[2] below). Floats are there to deal with the problem of the object that won't fit on the present page, and to help when you really don't want the object here just now.

Floats are not part of the normal stream of text, but separate entities, positioned in a part of the page to themselves (top, middle, bottom, left, right, or wherever the designer specifies). They always have a caption describing them and they are always numbered so they can be referred to from elsewhere in the text. LaTeX automatically floats Tables and Figures, depending on how much space is left on the page at the point that they are processed. If there is not enough room on the current page, the float is moved to the top of the next page. This can be changed by moving the Table or Figure definition to an earlier or later point in the text, or by adjusting some of the parameters which control automatic floating.

Authors sometimes have many floats occurring in rapid succession, which raises the problem of how they are supposed to fit on the page and still leave room for text. In this case, LaTeX stacks them all up and prints them together if possible, or leaves them to the end of the chapter in protest. The skill is to space them out within your text so that they intrude neither on the thread of your argument or discussion, nor on the visual balance of the typeset pages.

18.1.1. Figures

To create a figure that floats, use the `figure` environment.

```
\begin{figure}[placement specifier]
... figure contents ...
\end{figure}
```

1 Chapter 17 on page 211
2 Chapter 18.9 on page 244

The previous section mentioned how floats are used to allow LaTeX to handle figures, while maintaining the best possible presentation. However, there may be times when you disagree, and a typical example is with its positioning of figures. The *placement specifier* parameter exists as a compromise, and its purpose is to give the author a greater degree of control over where certain floats are placed.

Specifier	**Permission**
`h`	Place the float *here* , i.e., *approximately* at the same point it occurs in the source text (however, not *exactly* at the spot)
`t`	Position at the *top* of the page.
`b`	Position at the *bottom* of the page.
`p`	Put on a special *page* for floats only.
`!`	Override internal parameters LaTeX uses for determining "good" float positions.
`H`	Places the float at precisely the location in the LaTeX code. Requires the `float` package,[3] e.g., `\usepackage{float}`. This is somewhat equivalent to `h!`.

What you do with these *placement permissions* is to list which of the options you wish to make available to LaTeX. These are simply possibilities, and LaTeX will decide when typesetting your document which of your supplied specifiers it thinks is best. Frank Mittelbach describes the algorithm[4]:

- If a float is encountered, LaTeX attempts to place it immediately according to its rules (detailed later)
  - if this succeeds, the float is placed and that decision is never changed;
  - if this does not succeed, then LaTeX places the float into a holding queue to be reconsidered when the next page is started (but not earlier).
- Once a page has finished, LaTeX examines this holding queue and tries to empty it as best as possible. For this it will first try to generate as many float pages as possible (in the hope of getting floats off the queue). Once this possibility is exhausted, it will next try to place the remaining floats into top and bottom areas. It looks at all the remaining floats and either places them or defers them to a later page (i.e., re-adding them to the holding queue once more).
- After that, it starts processing document material for this page. In the process, it may encounter further floats.
- If the end of the document has been reached or if a \clearpage is encountered, LaTeX starts a new page, relaxes all restrictive float conditions, and outputs all floats in the holding queue by placing them on float page(s).

In some special cases LaTeX won't follow these positioning parameters and additional commands will be necessary, for example, if one needs to specify an alignment other than centered for a float that sits alone in one page[5].

3 `http://www.ctan.org/tex-archive/macros/latex/contrib/float/`
4 Float environment positioning, by Frank Mittelbach ^{`http://tex.stackexchange.com/a/39020`}
5 `http://tex.stackexchange.com/questions/28556/how-to-place-a-float-at-the-top-of-a-floats-only-page`

Use `\listoffigures` to add a list of the figures in the beginning of the document. To change the name used in the caption from **Figure** to **Example** , use `\renewcommand{\figurename}{Example}` in the figure contents.

Figures with borders

It's possible to get a thin border around all figures. You have to write the following once at the beginning of the document:

```
\usepackage{float}
\floatstyle{boxed}
\restylefloat{figure}
```

The border will not include the caption.

18.1.2. Tables

Floating tables are covered in a separate chapter[6]. Let's give a quick reminder here. The `tabular` environment that was used to construct the tables is not a float by default. Therefore, for tables you wish to float, wrap the `tabular` environment within a `table` environment, like this:

```
\begin{table}
  \begin{tabular}{...}
  ... table data ...
  \end{tabular}
\end{table}
```

You may feel that it is a bit long winded, but such distinctions are necessary, because you may not want all tables to be treated as a float.

Use `\listoftables` to add a list of the tables in the beginning of the document.

18.2. Keeping floats in their place

The `placeinshttp://www.ctan.org/pkg/placeins` package provides the command `\FloatBarrier`, which can be used to prevent floats from being moved over it. This can, e.g., be useful at the beginning of each section. The package even provides an option to change the definition of `\section` to automatically include a `\FloatBarrier`. This can be set by loading the package with the option `[section]` (`\usepackage[section]{placeins}`). `\FloatBarrier` may also be useful to prevent floats intruding on lists created using `itemize` or `enumerate`. The `flafter` package can be used to force floats to appear after they are defined, and the `endfloathttp://www.ctan.org/pkg/endfloat` package can be used to place all floats at the end of a document. The `floathttp://www.ctan.org/pkg/float` package provides the `H` option to floating environments, which stops them from floating.

6 Chapter 14.6 on page 173

18.3. Captions

It is always good practice to add a caption to any figure or table. Fortunately, this is very simple in LaTeX. All you need to do is use the `\caption{''text''}` command within the float environment. LaTeX will automatically keep track of the numbering of figures, so you do not need to include this within the caption text.

The location of the caption is traditionally underneath the float. However, it is up to you to therefore insert the caption command after the actual contents of the float (but still within the environment). If you place it before, then the caption will appear above the float. Try out the following example to demonstrate this effect:

```
\documentclass[a4paper,12pt]{article}

\usepackage[english]{babel}
\usepackage{graphicx}

\begin{document}

\begin{figure}[h!]
  \caption{A picture of a gull.}
  \centering
    \includegraphics[width=0.5\textwidth]{gull}
\end{figure}

\begin{figure}[h!]
  \centering
    \reflectbox{%
      \includegraphics[width=0.5\textwidth]{gull}<!---->}
  \caption{A picture of the same gull
           looking the other way!}
\end{figure}

\begin{table}[h!]
  \begin{center}
    \begin{tabular}{ l c r }
    \hline
    1 & 2 & 3 \\
    4 & 5 & 6 \\
    7 & 8 & 9 \\
    \hline
    \end{tabular}
  \end{center}
  \caption{A simple table}
\end{table}

Notice how the tables and figures
have independent counters.

\end{document}
```

Figure 1: A picture of a gull.

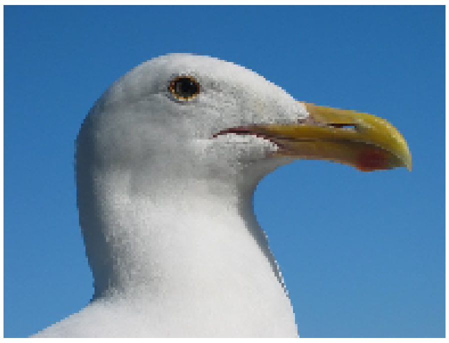


Figure 2: A picture of the same gull looking the other way!

1	2	3
4	5	6
7	8	9

Table 1: A simple table

Notice how the tables and figures have independent counters.

Figure 66

Note that the command `\reflectbox{...}` flips its content horizontally.

18.3.1. Side captions

It is sometimes desirable to have a caption appear on the side of a float, rather than above or below. The `sidecap` package can be used to place a caption beside a figure or table. The following example demonstrates this for a figure by using a `SCfigure` environment in place of the `figure` environment.

```
\documentclass{article}

\usepackage[pdftex]{graphicx}
\usepackage{sidecap}

\begin{document}

\begin{SCfigure}
  \centering
  \caption{ ... caption text ... }
  \includegraphics[width=0.3\textwidth]%
    {Giraff_picture}% picture filename
\end{SCfigure}

\end{document}
```

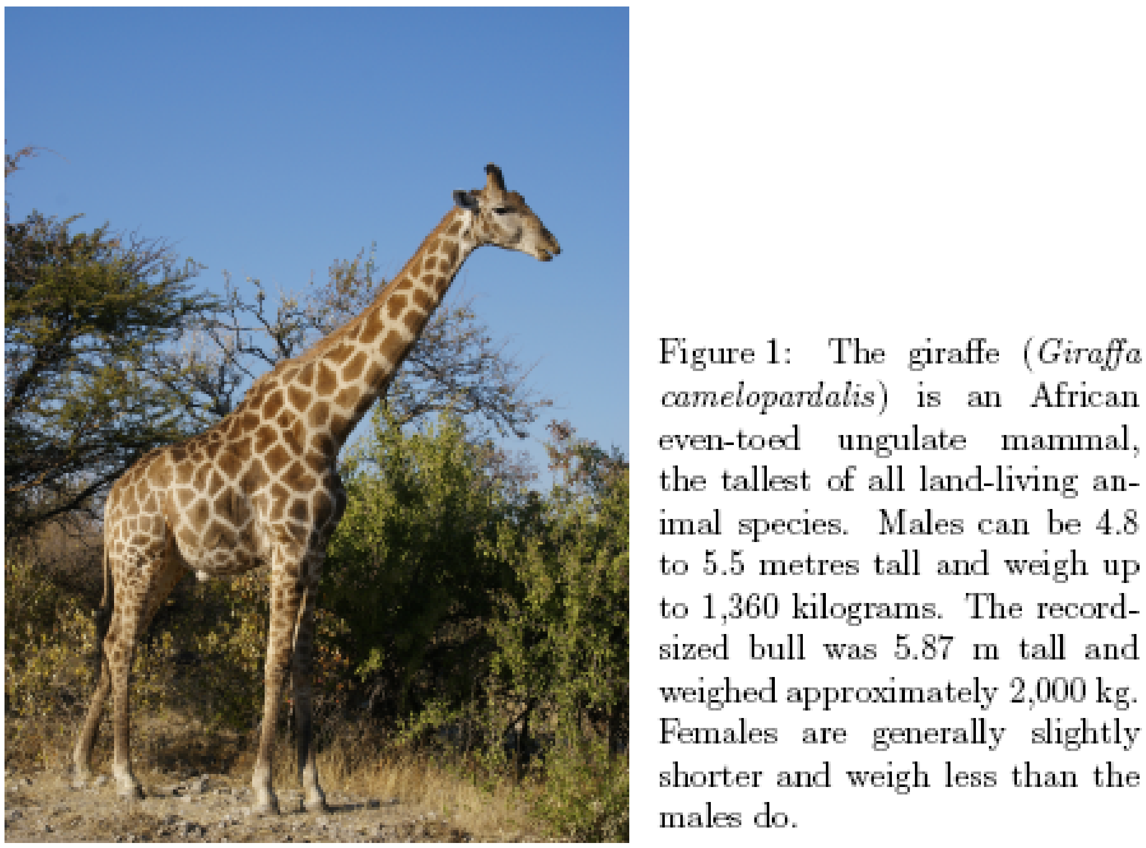


Figure 1: The giraffe (*Giraffa camelopardalis*) is an African even-toed ungulate mammal, the tallest of all land-living animal species. Males can be 4.8 to 5.5 metres tall and weigh up to 1,360 kilograms. The record-sized bull was 5.87 m tall and weighed approximately 2,000 kg. Females are generally slightly shorter and weigh less than the males do.

Figure 67

18.3.2. Unnumbered captions

In some types of document (such as presentations), it may not be desirable for figure captions to start *Figure:* . This is easy to suppress by just placing the caption text in the `Figure` environment, without enclosing it in a `Caption`. This however means that there is no caption available for inclusion in a list of figures.

18.3.3. Renaming table caption prefix

In case you want to rename your table caption from ”Table” to something else, you can use the `\captionsetup` command. For example,

```
\usepackage{caption}
\captionsetup[table]{name=New Table Name}
```

18.4. Lists of figures and tables

Captions can be listed at the beginning of a paper or report in a ”List of Tables” or a ”List of Figures” section by using the `\listoftables` or `\listoffigures` commands, respectively. The caption used for each figure will appear in these lists, along with the figure numbers, and page numbers that they appear on.

The `\caption` command also has an optional parameter, `\caption[''short'']{''long''}` which is used for the *List of Tables* or *List of Figures* . Typically the `short` description is for the caption listing, and the `long` description will be placed beside the figure or table. This is particularly useful if the caption is long, and only a ”one-liner” is desired in the figure/table listing. Here is an example of this usage:

```
\documentclass[12pt]{article}
\usepackage{graphicx}

\begin{document}

\listoffigures

\section{Introduction}

\begin{figure}[hb]
  \centering
  \includegraphics[width=4in]{gecko}
  \caption[Close up of \textit{Hemidactylus} sp.]
   {Close up of \textit{Hemidactylus} sp., which is
   part the genus of the gecko family. It is the
   second most speciose genus in the family.}
\end{figure}

\end{document}
```

List of Figures

1 Introduction

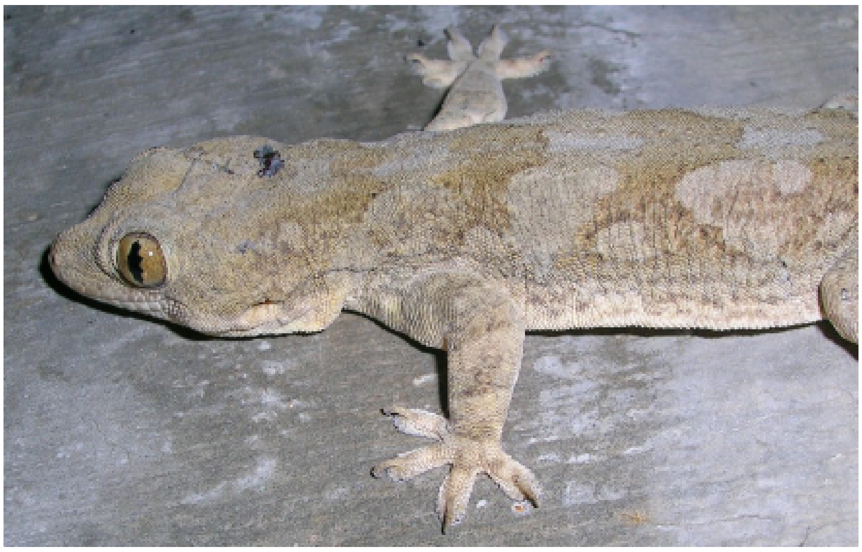


Figure 1: Close up of *Hemidactylus* sp., which is part the genus of the gecko family. It is the second most speciose genus in the family.

Figure 68

18.5. Labels and cross-referencing

Labels and cross-references work fairly similarly to the general case - see the Labels and Cross-referencing[7] section for more information.

> ⚠ **Warning**
>
> If you want to label a figure so that you can reference it later, you have to add the label **after the caption** (inside seems to work in LaTeX 2e) but **inside the floating environment** . If it is declared outside, it will give the section number.

If the label picks up the section or list number instead of the figure number, put the label inside the caption to ensure correct numbering. If you get an error when the label is inside the caption, use `\protect` in front of the `\label` command.

7 Chapter 21 on page 267

18.6. Wrapping text around figures

An author may prefer that some floats do not break the flow of text, but instead allow text to wrap around it. (Obviously, this effect only looks decent when the figure in question is significantly narrower than the text width.)

A word of warning: Wrapping figures in LaTex will require a lot of manual adjustment of your document. There are several packages available for the task, but none of them works perfectly. Before you make the choice of including figures with text wrapping in your document, make sure you have considered all the options. For example, you could use a layout with two columns for your documents and have no text-wrapping at all.

Anyway, we will look at the package `wrapfig`. Note that `wrapfig` may not come with the default installation of LaTeX; you might need to install additional packages[8]. Noted also, wrapfig is incompatible with the enumerate and itemize environments

To use `wrapfig`, you must first add this to the preamble:

```
\usepackage{wrapfig}
```

This then gives you access to:

```
\begin{wrapfigure}[lineheight]{position}{width}
```

There are overall eight possible positioning targets:

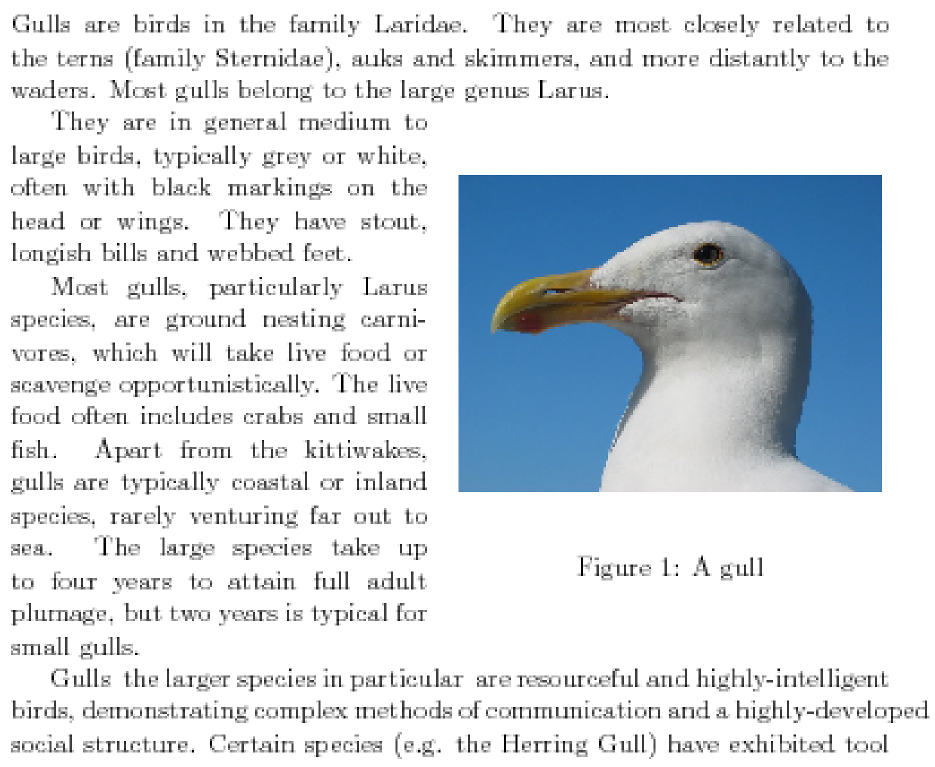
r	R	right side of the text
l	L	left side of the text
i	I	inside edge–near the binding (in a `twoside` document)
o	O	outside edge–far from the binding

The uppercase-character allows the figure to float, while the lowercase version means ”exactly here”. [9]

The *width* is, of course, the width of the figure. An example:

```
\begin{wrapfigure}{r}{0.5\textwidth}
  \begin{center}
    \includegraphics[width=0.48\textwidth]{gull}
  \end{center}
  \caption{A gull}
\end{wrapfigure}
```

8 Chapter 3 on page 29
9 `http://ftp.univie.ac.at/packages/tex/macros/latex/contrib/wrapfig/wrapfig-doc.pdf`



Gulls are birds in the family Laridae. They are most closely related to the terns (family Sternidae), auks and skimmers, and more distantly to the waders. Most gulls belong to the large genus Larus.

They are in general medium to large birds, typically grey or white, often with black markings on the head or wings. They have stout, longish bills and webbed feet.

Most gulls, particularly Larus species, are ground nesting carnivores, which will take live food or scavenge opportunistically. The live food often includes crabs and small fish. Apart from the kittiwakes, gulls are typically coastal or inland species, rarely venturing far out to sea. The large species take up to four years to attain full adult plumage, but two years is typical for small gulls.

Figure 1: A gull

Gulls the larger species in particular are resourceful and highly-intelligent birds, demonstrating complex methods of communication and a highly-developed social structure. Certain species (e.g. the Herring Gull) have exhibited tool

Figure 69

You can also allow LaTeX to assign a width to the wrap by setting the width to 0pt. \begin{wrapfigure}{l}{0pt}

Note that we have specified a size for both the `wrapfigure` environment and the image we have included. We did it in terms of the text width: it is always better to use relative sizes in LaTeX, let LaTeX do the work for you! The "wrap" is slightly bigger than the picture, so the compiler will not return any strange warning and you will have a small white frame between the image and the surrounding text. You can change it to get a better result, but if you don't keep the image smaller than the "wrap", you will see the image *over* the text.

The wrapfig package can also be used with user-defined floats with float package. See below in the section on custom floats[10].

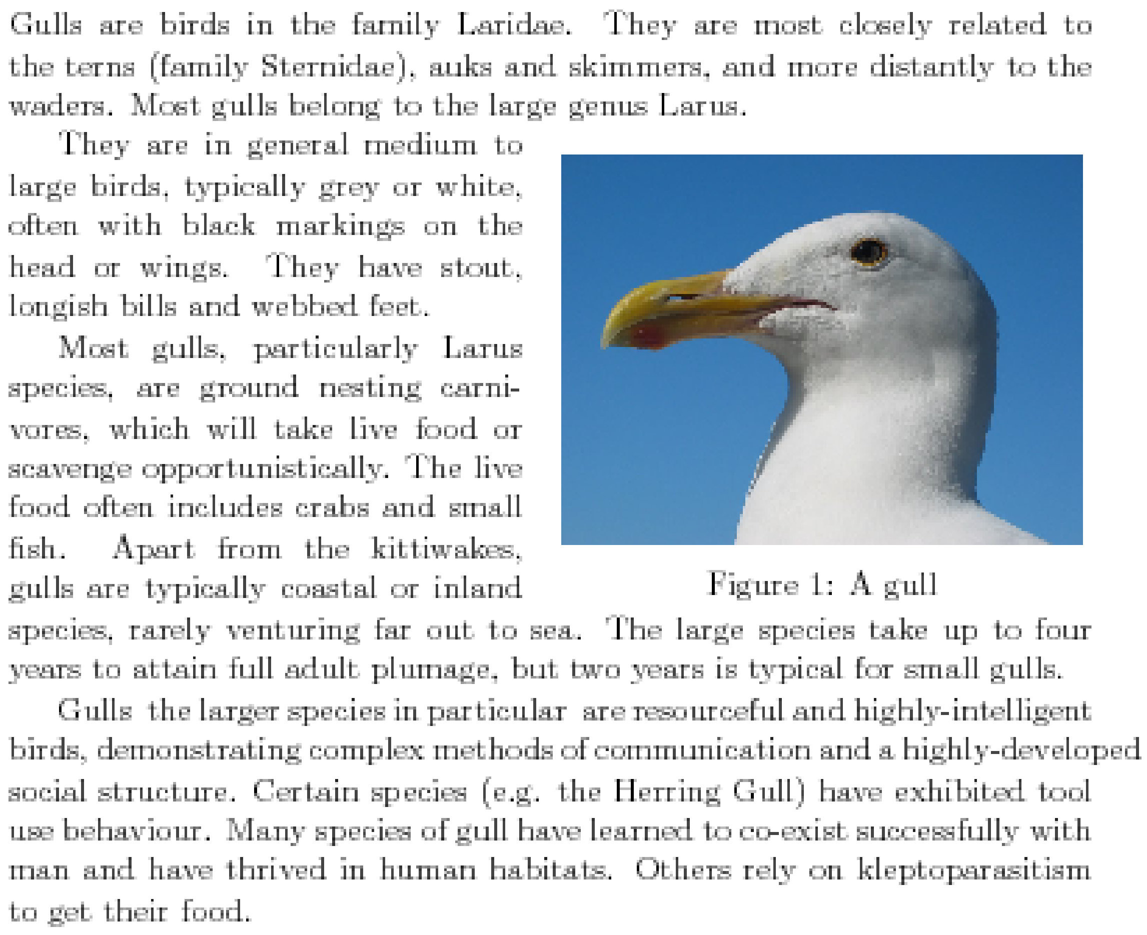
18.6.1. Tip for figures with too much white space

It happens that you'll generate figures with too much (or too little) white space on the top or bottom. In such a case, you can simply make use of the optional argument `[lineheight]`. It specifies the height of the figure in number of lines of text. Also remember that the environment `center` adds some extra white space at its top and bottom; consider using the command `\centering` instead.

10 Chapter 18.9 on page 244

Another possibility is adding space within the float using the `\vspace{...}` command. The argument is the size of the space you want to add, you can use any unit you want, including pt, mm, in, etc. If you provide a negative argument, it will add a *negative* space, thus removing some white space. Using `\vspace` tends to move the caption relative to the float while the `[lineheight]` argument does not. Here is an example using the `\vspace` command, the code is exactly the one of the previous case, we just added some negative vertical spaces to shrink everything up:

```
\begin{wrapfigure}{r}{0.5\textwidth}
  \vspace{-20pt}
  \begin{center}
    \includegraphics[width=0.48\textwidth]{gull}
  \end{center}
  \vspace{-20pt}
  \caption{A gull}
  \vspace{-10pt}
\end{wrapfigure}
```


Gulls are birds in the family Laridae. They are most closely related to the terns (family Sternidae), auks and skimmers, and more distantly to the waders. Most gulls belong to the large genus Larus.

They are in general medium to large birds, typically grey or white, often with black markings on the head or wings. They have stout, longish bills and webbed feet.

Most gulls, particularly Larus species, are ground nesting carnivores, which will take live food or scavenge opportunistically. The live food often includes crabs and small fish. Apart from the kittiwakes, gulls are typically coastal or inland species, rarely venturing far out to sea. The large species take up to four years to attain full adult plumage, but two years is typical for small gulls.

Figure 1: A gull

Gulls the larger species in particular are resourceful and highly-intelligent birds, demonstrating complex methods of communication and a highly-developed social structure. Certain species (e.g. the Herring Gull) have exhibited tool use behaviour. Many species of gull have learned to co-exist successfully with man and have thrived in human habitats. Others rely on kleptoparasitism to get their food.

Figure 70 336

In this case it may look too shrunk, but you can manage spaces the way you like. In general, it is best not to add any space at all: let LaTeX do the formatting work!

(In this case, the problem is the use of `\begin{center}` to center the image. The `center` environment adds extra space that can be avoided if `\centering` is used instead.)

You can use `intextsep` parameter to control additional space above and below the figure: `\setlength\intextsep{0pt}`

Alternatively you might use the `picins` package instead of the wrapfig package which produces a correct version without the excess white space out of the box without any hand tuning.

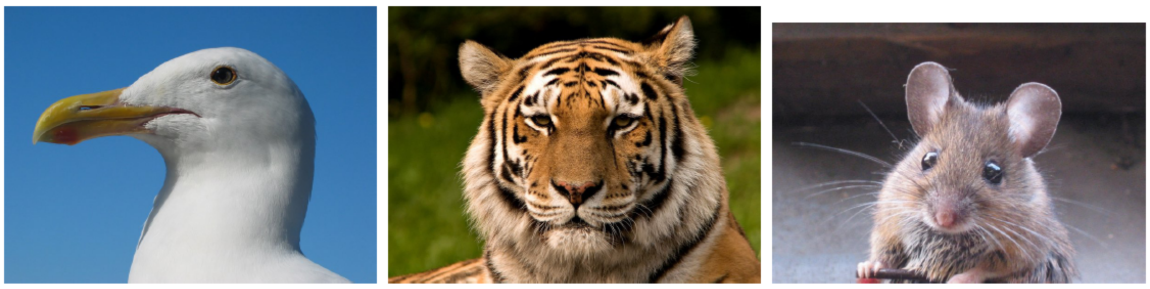
There is also an alternative to `wrapfig`: the package `floatflt` `http://www.ctan.org/pkg/floatflt`.

To remove the white space from a figure once for all, one should refer to the program pdfcrop, included in most TeX installations.

18.7. Subfloats

A useful extension is the `subcaptionhttp://www.ctan.org/pkg/subcaption` package which uses subfloats within a single float. The `subfigure` and `subfig` packages are deprecated however they are useful alternatives when used in-conjunction with latex templates (i.e templates for journals from Springer and IOP, IEEETran and ACM SIG) that are not compatible with `subcaption`. These packages give the author the ability to have subfigures within figures, or subtables within table floats. Subfloats have their own caption, and an optional global caption. An example will best illustrate the usage of the `subcaption` package:

```
\usepackage{graphicx}
\usepackage{caption}
\usepackage{subcaption}

\begin{figure}
        \centering
        \begin{subfigure}[b]{0.3\textwidth}
                \includegraphics[width=\textwidth]{gull}
                \caption{A gull}
                \label{fig:gull}
        \end{subfigure}%
        ~ %add desired spacing between images, e. g. ~, \quad, \qquad, \hfill etc.
          %(or a blank line to force the subfigure onto a new line)
        \begin{subfigure}[b]{0.3\textwidth}
                \includegraphics[width=\textwidth]{tiger}
                \caption{A tiger}
                \label{fig:tiger}
        \end{subfigure}
        ~ %add desired spacing between images, e. g. ~, \quad, \qquad, \hfill etc.
          %(or a blank line to force the subfigure onto a new line)
        \begin{subfigure}[b]{0.3\textwidth}
                \includegraphics[width=\textwidth]{mouse}
                \caption{A mouse}
                \label{fig:mouse}
        \end{subfigure}
        \caption{Pictures of animals}\label{fig:animals}
\end{figure}
```



(a) A gull (b) A tiger (c) A mouse

Figure 1: Pictures of animals

Figure 71

You will notice that the figure environment is set up as usual. You may also use a table environment for subtables. For each subfloat, you need to use:

```
\begin{table}[<placement specifier>]
    \begin{subtable}[<placement specifier>]{<width>}
        \centering
        ... table 1 ...
        \caption{<sub caption>}
    \end{subtable}
    ~
    \begin{subtable}[<placement specifier>]{<width>}
        \centering
        ... table 2 ...
        \caption{<sub caption>}
    \end{subtable}
\end{table}
```

If you intend to cross-reference any of the subfloats, see where the label is inserted; `\caption` outside the `subfigure`-environment will provide the global caption.

`subcaption` will arrange the figures or tables side-by-side providing they can fit, otherwise, it will automatically shift subfloats below. This effect can be added manually, by putting the newline command (\\) before the figure you wish to move to a newline.

Horizontal spaces between figures are controlled by one of several commands, which are placed in between `\begin{subfigure}` and `\end{subfigure}`:

- A non-breaking space (specified by ~ as in the example above) can be used to insert a space in between the subfigs.
- Math spaces[11]: `\qquad`, `\quad`, `\;`, and `\,`
- Generic space: `\hspace{''length''}`
- Automatically expanding/contracting space: `\hfill`

11 Chapter 27.15 on page 329

18.8. Wide figures in two column documents

If you are writing a document using two columns (i.e. you started your document with something like `\documentclass[twocolumn]{article}`), you might have noticed that you can't use floating elements that are wider than the width of a column (using a LaTeX notation, wider than `0.5\textwidth`), otherwise you will see the image overlapping with text. If you really have to use such wide elements, the only solution is to use the "starred" variants of the floating environments, that are `{figure*}` and `{table*}`. Those "starred" versions work like the standard ones, but they will be as wide as the page, so you will get no overlapping.

A bad point of those environments is that they can be placed only at the top of the page or on their own page. If you try to specify their position using modifiers like *b* or *h* they will be ignored. Add `\usepackage{dblfloatfix}` to the preamble in order to alleviate this problem with regard to placing these floats at the bottom of a page, using the optional specifier `[b]`. Default is `[tbp]`. However, *h* still does not work.

To prevent the figures from being placed out-of-order with respect to their "non-starred" counterparts, the package `fixltx2e` [12] should be used (e.g. `\usepackage{fixltx2e}`).

18.9. Custom floats

If tables and figures are not adequate for your needs, then you always have the option to create your own! Examples of such instances could be source code examples, or maps. For a program float example, one might therefore wish to create a float named `program`. The package `float` is your friend for this task. *All commands to set up the new float must be placed in the preamble, and not within the document.*

1. Add `\usepackage{float}` to the preamble of your document
2. Declare your new float using: `\newfloat{type}{placement}{ext}[outer counter]`, where:
   - *type* - the new name you wish to call your float, in this instance, 'program'.
   - *placement* - t, b, p, or h (as previously described in Placement[13]), where letters enumerate permitted placements.
   - *ext* - the file name extension of an auxiliary file for the list of figures (or whatever). Latex writes the captions to this file.
   - *outer counter* - the presence of this parameter indicates that the counter associated with this new float should depend on outer counter, for example 'chapter'.
3. The default name that appears at the start of the caption is the type. If you wish to alter this, use `\floatname{type}{floatname}`
4. Changing float style can be issued with `\floatstyle{style}` (Works on all subsequent `\newfloat` commands, therefore, must be inserted before `\newfloat` to be effective).
   - `plain` - the normal style for Latex floats, but the caption is always below the content.

12 `http://www.tex.ac.uk/cgi-bin/texfaq2html?label=2colfltorder`
13 Chapter 18.6 on page 239

- `plaintop` - the normal style for Latex floats, but the caption is always above the content.
- `boxed` - a box is drawn that surrounds the float, and the caption is printed below.
- `ruled` - the caption appears above the float, with rules immediately above and below. Then the float contents, followed by a final horizontal rule.

Float styles can also be customized as the second example below illustrates.

An example document using a new `program` float type:

```
\documentclass{article}

\usepackage{float}

\floatstyle{ruled}
\newfloat{program}{thp}{lop}
\floatname{program}{Program}

\begin{document}

\begin{program}
  \begin{verbatim}

class HelloWorldApp {
  public static void main(String[] args) {
    //Display the string
    System.out.println("Hello World!");
  }
}
\end{verbatim}
  \caption{The Hello World! program in Java.}
\end{program}

\end{document}
```

The `verbatim` environment is an environment that is already part of Latex. Although not introduced so far, its name is fairly intuitive! LaTeX will reproduce everything you give it, including new lines, spaces, etc. It is good for source code, but if you want to introduce a lot of code you might consider using the `listings` package, that was made just for it.

While this is useful, one should be careful when embedding the float within another float. In particular, the error

```
not in outer par mode
```

may occur. One solution might be to use the [H] option (not any other) on the inner float, as this option ”pins” the inner float to the outer one.

Newly created floats with `\newfloat` can also be used in combination with the `wrapfig` package from above. E.g. the following code creates a floating text box, which floats in the text on the right side of the page and is complete with caption, numbering, an index file with the extension .lob and a customization of the float’s visual layout:

```
\documentclass{article}

% have hyperref package before float in order to get strange errors with
 .\theHfloatbox
\usepackage[pdftex]{hyperref}
```

```
\usepackage{float}

% allows use of "@" in control sequence names
\makeatletter

% this creates a custom and simpler ruled box style
\newcommand\floatc@simplerule[2]{{\@fs@cfont #1 #2}\par}
\n
ewcommand\fs@simplerule{\def\@fs@cfont{\bfseries}\let\@fs@capt\floatc@simplerule
  \def\@fs@pre{\hrule height.8pt depth0pt \kern4pt}%
  \def\@fs@post{\kern4pt\hrule height.8pt depth0pt \kern4pt \relax}%
  \def\@fs@mid{\kern8pt}%
  \let\@fs@iftopcapt\iftrue}

% this code block defines the new and custom floatbox float environment
\floatstyle{simplerule}
\newfloat{floatbox}{thp}{lob}[section]
\floatname{floatbox}{Text Box}

\begin{document}

\begin{floatbox}{r}{}
  \textit{Bootstrapping} is a resampling technique used
  for robustly estimating statistical quantities, such as
  the model fit $R^2$. It offers some protection against
  the sampling bias.
  \caption{Bootstrapping}
\end{floatbox}

\end{document}
```

18.9.1. Caption styles

To change the appearance of captions, use the `caption` `http://www.ctan.org/pkg/caption` package. For example, to make all caption labels small and bold:

```
\usepackage[font=small,labelfont=bf]{caption}
```

The KOMA script packages `http://www.komascript.de/` have their own caption customizing features with e.g. `\captionabove`, `\captionformat` and `\setcapwidth`. However these definitions have limited effect on newly created float environments with the `wrapfig` package.

Alternatively, you can redefine the `\thefigure` command:

```
\renewcommand{\thefigure}{\arabic{section}.\arabic{figure}}
```

See this page[14] for more information on counters. Finally, note that the `caption2` package has long been deprecated.

14 Chapter 24 on page 291

18.10. Labels in the figures

There is a LaTeX package `lpic` `http://www.ctan.org/pkg/lpic` to put LaTeX on top of included graphics, thus allowing to add TeX annotations to imported graphics. It defines a convenient interface to put TeX over included graphics, and allows for drawing a white background under the typeset material to overshadow the graphics. It is a better alternative for labels inside of graphics; you do not have to change text size when rescaling pictures, and all LaTeX power is available for labels.

A very similar package, with somewhat different syntax, is `pinlabel` `http://www.ctan.org/pkg/pinlabel`. The link given also points to the packages `psfrag` and `overpic`.

A much more complicated package which can be used in the same way is TikZ[15]. TikZ is a front-end to a drawing library called pgf (which is used to make beamer for instance). It can be used to label figures by adding text nodes on top of an image node.

18.11. Summary

That concludes all the fundamentals of floats. You will hopefully see how much easier it is to let LaTeX do all the hard work and tweak the page layouts in order to get your figures in the best place. As always, the fact that LaTeX takes care of all caption and reference numbering is a great time saver.

18.12. Notes and references

This page uses material from Andy Roberts' Getting to grips with LaTeX with permission from the author.

15 Chapter 47 on page 535

19. Footnotes and Margin Notes

19.1. Footnotes

Footnotes are a very useful way of providing extra information to the reader. Usually, it is non-essential information which can be placed at the bottom of the page. This keeps the main body of text concise.

The footnote facility is easy to use. The command you need is:

```
\footnote{text}
```

. Do not leave a space between the command and the word where you wish the footnote marker to appear, otherwise LaTeX will process that space and will leave the output not looking as intended.

```
Creating a footnote is easy.\footnote{An example footnote.}
```

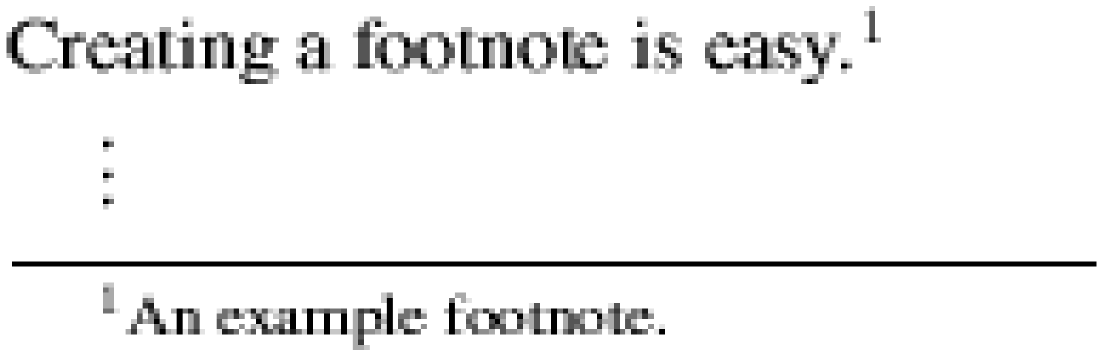


Figure 72

LaTeX will obviously take care of typesetting the footnote at the bottom of the page. Each footnote is numbered sequentially - a process that, as you should have guessed by now, is automatically done for you.

You can also choose to place the footnote text manually. In this case we use the `\footnotemark`-`\footnotetext` duo:

```
\footnotemark
% ...
Somewhere else\footnotetext{This is my footnote!}
```

The footnote number can also be explicitly specified.

```
\footnotemark[17]
% ...
Somewhere else\footnotetext[17]{This is my footnote!}
```

19.1.1. Customization

It is possible to customize the footnote marking. By default, they are numbered sequentially (Arabic). However, without going too much into the mechanics of LaTeX at this point, it is possible to change this using the following command (which needs to be placed at the beginning of the document, or at least before the first footnote command is issued).

`\renewcommand{\thefootnote}{\arabic{footnote}}`	Arabic numerals, e.g., 1, 2, 3...
`\renewcommand{\thefootnote}{\roman{footnote}}`	Roman numerals (lowercase), e.g., i, ii, iii...
`\renewcommand{\thefootnote}{\Roman{footnote}}`	Roman numerals (uppercase), e.g., I, II, III...
`\renewcommand{\thefootnote}{\alph{footnote}}`	Alphabetic (lowercase), e.g., a, b, c...
`\renewcommand{\thefootnote}{\Alph{footnote}}`	Alphabetic (uppercase), e.g., A, B, C...
`\renewcommand{\thefootnote}{\fnsymbol{footnote}}`	A sequence of nine symbols, try it and see!

To make a footnote without number mark use this declaration:

```
\let\thefootnote\relax\footnote{There is no number in this footnote}
```

In this way, the numbering is switched off globally. To have only one footnote without number mark, the above command has to be placed between { }. Nevertheless, in that case, the current footnote counter is still incremented, so for instance you'd get footnote 1, unnumbered, and 2. A better solution[1] consists in defining the following macro in the preamble, and to use it:

```
\makeatletter
\def\blfootnote{\xdef\@thefnmark{}\@footnotetext}
\makeatother
```

The package footmisc[3] offers many possibilities for customizing the appearance of footnotes. It can be used, for example, to use a different font within footnotes.

19.1.2. Reset counter

every section

```
\makeatletter
```

1 LaTeX footnotes [2]. . Retrieved 2011-09-30
3 `http://www.ctan.org/pkg/footmisc`

```
\@addtoreset{footnote}{section}
\makeatother
```

every page

(This may require running LaTeX twice)

```
\usepackage{perpage} %the perpage package
\MakePerPage{footnote} %the perpage package command
```

19.1.3. Common problems and workarounds

- Footnotes unfortunately don't work with tables, as it is considered a bad practice. You can overcome this limitation with several techniques: you can use `\footnotemark[123]` in the table, and `\footnotetext[123]{HelloWorld!}` somewhere on the page. The same with references: use `\footnote{HelloWorld!\label{fnote}}` somewhere on the page and `\textsuperscript{\ref{fnote}}` in the table. Or, you can add `\usepackage{footnote}` and `\makesavenoteenv{tabular}` to the preamble, and put your `table` environment in a `\begin{savenotes}` environment. Note that the latter does not work with the packages `color` or `colortbl`. See this FAQ page[4] for other approaches (such as the use of tablenotes with `threeparttable`).
- Footnotes also don't work inside minipage environment (In fact, several environments break footnote support. the `\makesavenoteenv{environmentname}` command of the footnote package might fix most). The minipage includes its own footnotes, independent of the document's. The package mpfnmark[5] allows greater flexibility in managing these two sets of footnotes.
- If the text within the footnote is a URL (using `\url` or `\href` commands) with special characters, it will not compile. You must either escape the characters with a leading backslash, or use another command.
- If the text within the footnote is very long, LaTeX may split the footnote over several pages. You can prevent LaTeX from doing so by increasing the penalty for such an operation. To do this, insert the following line into the preamble of your document:

```
\interfootnotelinepenalty=10000
```

- To make multiple references to the same footnote, you can use the following syntax:

```
Text that has a footnote\footnote{This is the footnote} looks like this. Later
 text referring to same footnote\footnotemark[\value{footnote}] uses the other
 command.
```

If you need hyperref support, use instead:

```
Text that has a footnote\footnote{This is the
 footnote}\addtocounter{footnote}{-1}\addtocounter{Hfootnote}{-1} looks like
 this. Later text referring to same footnote\footnotemark uses the other command.
```

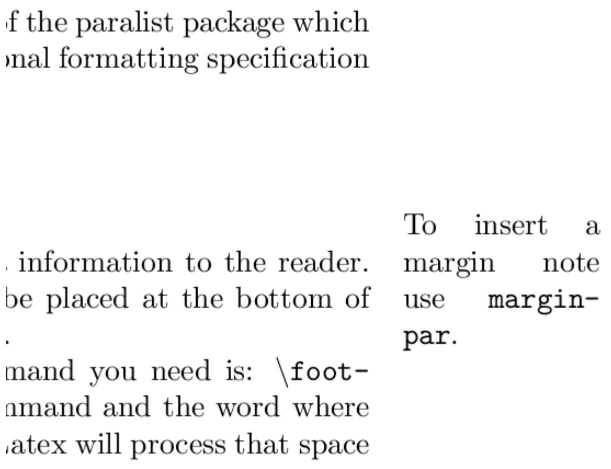
4 `http://www.tex.ac.uk/cgi-bin/texfaq2html?label=footintab`

5 `http://www.cs.brown.edu/system/software/latex/doc/mpfnmark.pdf`

Note that these approaches will not work if there are other footnotes between the first reference and the subsequent "duplicate" references. For more general solutions, see here[6] and here[7].

- If the footnote is intended to be added to the title of a chapter, a section, or similar, two methods can be used:
  1. Write `\section[title] {title\footnote{I'm a footnote referred to the section} }` where `title` is the title of the section.
  2. Use the `footmisc` package, with package option `stable`, and simply add the footnote to the section title.

19.2. Margin Notes



f the paralist package which
nal formatting specification

information to the reader.
be placed at the bottom of
.
mand you need is: \foot-
nmand and the word where
atex will process that space

To insert a margin note use margin-par.

Figure 73 A margin note.

Margin Notes are useful during the editorial process, to exchange comments among authors. To insert a margin note use `\marginpar{margin text}`. For one-sided layout (simplex), the text will be placed in the right margin, starting from the line where it is defined. For two-sided layout (duplex), it will be placed in the outside margin and for two-column layout it will be placed in the nearest margin.

To swap the default side, use `\reversemarginpar` and margin notes will then be placed on the opposite side, which would be the inside margin for two-sided layout.

If the text of your marginpar depends on which margin it is put in (say it includes an arrow pointing at the text or refers to a direction as in "as seen to the left..."), you can use `\marginpar[left text]{right text}` to specify the variants.

6 http://tex.stackexchange.com/questions/35043
7 http://tex.stackexchange.com/questions/10102/multiple-references-to-the-same-footnote-with-hyperref-sup

To insert a margin note in an area that `\marginpar` can't handle, such as footnotes or equation environments, use the package `marginnote`.

Another option for adding colored margin notes in a fancy way provides the package `todonotes` by using `\todo{todo note}`. It makes use of the package `pgf` used for designing and drawing with a huge tool database.

The packages `mparhack` and `marginnote` can be used if the native `\marginpar` command does not meet your needs.

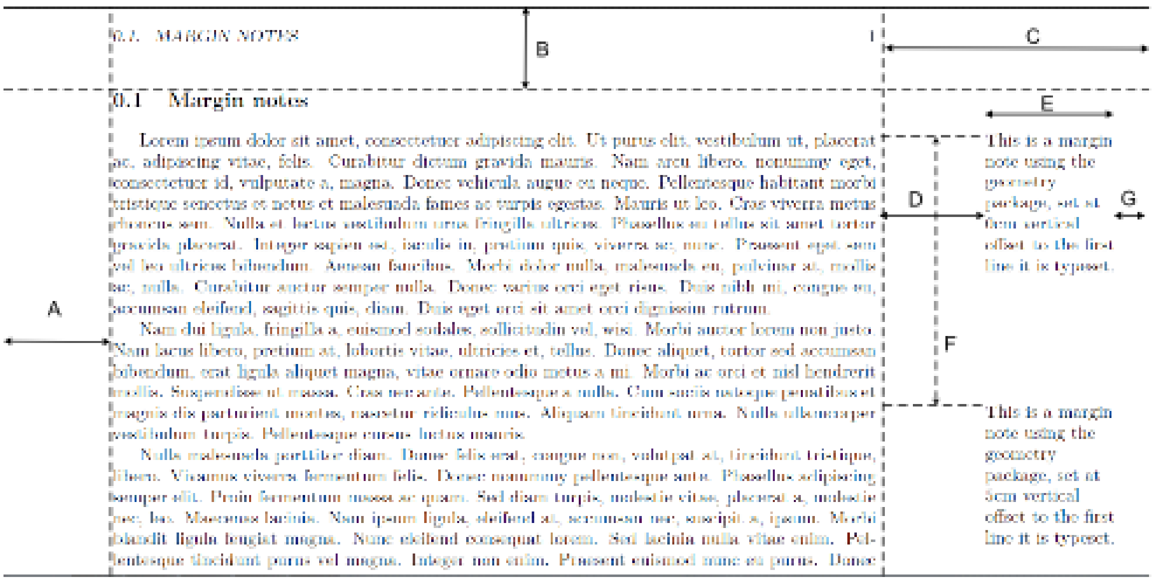

0.1. MARGIN NOTES
B
C
0.1 Margin notes
E
This is a margin note using the geometry package, set at 0cm vertical offset to the first line it is typeset.
D
G
A
F
This is a margin note using the geometry package, set at 5cm vertical offset to the first line it is typeset.

Figure 74 Margin geometry (bottom margin H not shown).

The marginnote and geometry package can set the widths of the margins and marginnotes as follows.

In the preamble, insert

```
\usepackage{marginnote}
```

and use the geometry package with custom sizes:

```
\usepackage[top=Bcm, bottom=Hcm, outer=Ccm, inner=Acm, heightrounded,
 marginparwidth=Ecm, marginparsep=Dcm]{geometry}
```

where A, B, C, D, E, F, G, X are all numbers in cm (of course other units than cm can be used).

In the main text, employ the marginnote package according to:

```
\marginnote{typeset text here...}[Fcm]
```

Specifically,

- `marginparwidth` (E) is the width of the margin note,
- `marginparsep` (D) is the separation between the paragraph and the margin note,

- F is the downwards vertical offset from the first line the margin note was written (negative values of F shift the margin note upwards), and
- the value *G = C – (D + E)* is the separation between the edge of the margin note and the edge.

The example on the right was typeset by the following:

```
\documentclass[a4paper,twoside,english]{article}
\usepackage{lmodern}
\renewcommand{\sfdefault}{lmss}
\usepackage[T1]{fontenc}

\makeatletter
\special{papersize=\the\paperwidth,\the\paperheight}

\usepackage{lipsum}
\usepackage{marginnote}
\usepackage[top=1.5cm, bottom=1.5cm, outer=5cm, inner=2cm, heightrounded,
 marginparwidth=2.5cm, marginparsep=2cm]{geometry}

\makeatother

\usepackage{babel}
\begin{document}

\section{Margin notes}

\marginnote{This is a margin note using the geometry package, set at 0cm
 vertical offset to the first line it is typeset.}[0cm]
\marginnote{This is a margin note using the geometry package, set at 5cm
 vertical offset to the first line it is typeset.}[5cm]
\lipsum[1-10]
\end{document}
```

19.3. Notes and References

This page uses material from Andy Roberts' Getting to grips with LaTeX with permission from the author.

ru:LaTeX/Подстрочные примечания и заметки на полях[8]

[8] http://ru.wikibooks.org/wiki/LaTeX%2F%D0%9F%D0%BE%D0%B4%D1%81%D1%82%D1%80%D0%BE%D1%87%D0%BD%D1%8B%D0%B5%20%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%87%D0%B0%D0%BD%D0%B8%D1%8F%20%D0%B8%20%D0%B7%D0%B0%D0%BC%D0%B5%D1%82%D0%BA%D0%B8%20%D0%BD%D0%B0%20%D0%BF%D0%BE%D0%BB%D1%8F%D1%85

20. Hyperlinks

LaTeX enables typesetting of hyperlinks, useful when the resulting format is PDF, and the hyperlinks can be followed. It does so using the package `hyperref`.

20.1. Hyperref

The package `hyperref`[1] provides LaTeX the ability to create hyperlinks within the document. It works with *pdflatex* and also with standard "latex" used with dvips and ghostscript or dvipdfm to build a PDF file. If you load it, you will have the possibility to include interactive external links and all your internal references will be turned to hyperlinks. The compiler *pdflatex* makes it possible to create PDF files directly from the LaTeX source, and PDF supports more features than DVI. In particular PDF supports hyperlinks, and the only way to introduce them in LaTeX is using `hyperref`. Moreover, PDF can contain other information about a document such as the title, the author, etc., which can be edited using this same package.

20.2. Usage

The basic usage with the standard settings is straightforward. Just load the package in the preamble:

```
\usepackage{hyperref}
```

This will automatically turn all your internal references into hyperlinks. It won't affect the way to write your documents: just keep on using the standard `\label`-`\ref` system (discussed in the chapter on Labels and Cross-referencing[2]); with `hyperref` those "connections" will become links and you will be able to click on them to be redirected to the right page. Moreover the table of contents, list of figures/tables and index will be made of hyperlinks, too. The hyperlinks will not show-up if you are working in draft mode.

20.2.1. Commands

The package provides some useful commands for inserting links pointing outside the document.

1 Hyperref package webpage in CTAN ^{`http://www.ctan.org/tex-archive/macros/latex/contrib/hyperref`}

2 Chapter 21 on page 267

\hyperref

Usage:

```
\hyperref[label_name]{''link text''}
```

This will have the same effect as `\ref{label_name}` but will make the text *link text* a full link, instead. The two can be combined. If the lemma labelled as `mainlemma` was number 4.1.1 the following example would result in

```
We use \hyperref[mainlemma]{lemma \ref*{mainlemma} }.
```

We use lemma 4.1.1.

with the hyperlink as expected. Note the "*" after `\ref` for avoiding nested hyperlinks.

\url

Usage:

```
\url{<my_url>}
```

It will show the URL using a mono-spaced font and, if you click on it, your browser will be opened pointing at it.

\href

Usage:

```
\href{<my_url>}{<description>}
```

It will show the string `description` using standard document font but, if you click on it, your browser will be opened pointing at `my_url`. Here is an example:

```
\url{http://www.wikibooks.org}
\href{http://www.wikibooks.org}{Wikibooks home}
```

Both point at the same page, but in the first case the URL will be shown, while in the second case the URL will be hidden. Note that, if you print your document, the link stored using `\href` will not be shown anywhere in the document.

20.2.2. Other possibilities

Apart from linking to websites discussed above, `hyperref` can be used to provide *mailto* links, links to local files, and links to anywhere within the PDF output file.

E-mail address

A possible way to insert email links is by

```
\href{mailto:my_address@wikibooks.org}{my\_address@wikibooks.org}
```

It just shows your email address (so people can know it even if the document is printed on paper) but, if the reader clicks on it, (s)he can easily send you an email. Or, to incorporate the `url` package's formatting and line breaking abilities into the displayed text, use[3]

```
\href{mailto:my_address@wikibooks.org}{\nolinkurl{my_address@wikibooks.org} }
```

When using this form, note that the `\nolinkurl` command is fragile and if the hyperlink is inside of a moving argument, it must be preceeded by a `\protect` command.

Local file

Files can also be linked using the url or the href commands. You simply have to add the string `run:` at the beginning of the link string:

```
\url{run:/path/to/my/file.ext}
\href{run:/path/to/my/file.ext}{text displayed}
```

Following `http://tex.stackexchange.com/questions/46488/link-to-local-pdf-file` the version with `url` does not always work, but `href` does.

It is possible to use relative paths to link documents near the location of your current document; in order to do so, use the standard Unix-like notation (`./` is the current directory, `../` is the previous directory, etc.)

Hyperlink and Hypertarget

It is also possible to create an anchor anywhere in the document (with or without caption) and to link to it. To create an anchor, use:

```
\hypertarget{label}{target caption}
```

and to link to it, use:

```
\hyperlink{label}{link caption}
```

where the `target caption` and `link caption` are the text that is displayed at the target location and link location respectively.

3 Email link with hyperref, url packages [4]. . Retrieved

20.3. Customization

The standard settings should be fine for most users, but if you want to change something, that is also possible. There are several variables and two methods to pass those to the package. Options can be passed as an argument of the package when it is loaded (the standard way packages work), or the `\hypersetup` command can be used as follows:

```
\hypersetup{<option1> [, ...]}
```

you can pass as many options as you want; separate them with a comma. Options have to be in the form:

```
variable_name=new_value
```

exactly the same format has to be used if you pass those options to the package while loading it, like this:

```
\usepackage[<option1, option2>]{hyperref}
```

Here is a list of the possible variables you can change (for the complete list, see the official documentation). The default values are written in an upright font:

Checkout 3.8 Big list at hyperref-manual at tug.org [5]

variable	**values**	**comment**
`bookmarks`	`=true,`*`false`*	show or hide the bookmarks bar when displaying the document
`unicode`	`=false,`*`true`*	allows to use characters of non-Latin based languages in Acrobat's bookmarks

5 `http://www.tug.org/applications/hyperref/manual.html#x1-120003.8`

variable	**values**	**comment**
`pdfborder`	`={`*`RadiusH RadiusV Width`* `[`*`Dash-Pattern`* `]}`	set the style of the border around a link. The first two parameters (RadiusH, RadiusV) have no effect in most pdf viewers. *Width* defines the thickness of the border. *Dash-Pattern* is a series of numbers separated by space and enclosed by box-brackets. It is an optional parameter to specify the length of each line & gap in the dash pattern. For example, {0 0 0.5 [3 3]} is supposed to draw a square box (no rounded corners) of width 0.5 and a dash pattern with a dash of length 3 followed by a gap of length 3. There is no uniformity in whether/how different pdf viewers render the dash pattern.
`pdftoolbar`	`=true,`*`false`*	show or hide Acrobat's toolbar
`pdfmenubar`	`=true,`*`false`*	show or hide Acrobat's menu
`pdffitwindow`	`=true,`*`false`*	resize document window to fit document size
`pdfstartview`	`={FitH},`*`{FitV}`* `,etc`[6]`.`	fit the width of the page to the window
`pdftitle`	`={text}`	define the title that gets displayed in the "Document Info" window of Acrobat
`pdfauthor`	`={text}`	the name of the PDF's author, it works like the one above
`pdfsubject`	`={text}`	subject of the document, it works like the one above
`pdfcreator`	`={text}`	creator of the document, it works like the one above
`pdfproducer`	`={text}`	producer of the document, it works like the one above
`pdfkeywords`	`={text}`	list of keywords, separated by brackets, example below

6 Other possible values are defined in the hyperref manual ^`{http://mirror.switch.ch/ftp/mirror/tex/macros/latex/contrib/hyperref/doc/manual.html#TBL-7-40-1}`

variable	**values**	**comment**
`pdfnewwindow`	`(=true,`*`false`*`)`	define if a new PDF window should get opened when a link leads out of the current document. NB: This option is ignored if the link leads to an http/https address.
`pagebackref`	`(=false,`*`true`*`)`	activate back references inside bibliography. Must be specified as part of the \usepackage{} statement.
`colorlinks`	`(=false,`*`true`*`)`	surround the links by color frames (`false`) or colors the text of the links (`true`). The color of these links can be configured using the following options (default colors are shown):
`hidelinks`		hide links (removing color and border)
`linkcolor`	`=red`	color of internal links (sections, pages, etc.)
`linktoc`	`=`*`none`*` ,section,`*`page`*` ,`*`all`*	defines which part of an entry in the table of contents is made into a link
`citecolor`	`=green`	color of citation links (bibliography)
`filecolor`	`=cyan`	color of file links
`urlcolor`	`=magenta`	color of URL links (mail, web)
`linkbordercolor`	`={1 0 0}`	color of frame around internal links (if `colorlinks=false`)
`citebordercolor`	`={0 1 0}`	color of frame around citations
`urlbordercolor`	`={0 1 1}`	color of frame around URL links

Please note, that explicit RGB specification is only allowed for the border colors (like linkbordercolor etc.), while the others may only assigned to named colors (which you can define your own, see Colors[7]). In order to speed up your customization process, here is a list with the variables with their default value. Copy it in your document and make the changes you want. Next to the variables, there is a short explanations of their meaning:

```
\hypersetup{
    bookmarks=true,          % show bookmarks bar?
```

7 Chapter 8 on page 89

```
    unicode=false,          % non-Latin characters in Acrobat's bookmarks
    pdftoolbar=true,        % show Acrobat's toolbar?
    pdfmenubar=true,        % show Acrobat's menu?
    pdffitwindow=false,     % window fit to page when opened
    pdfstartview={FitH},    % fits the width of the page to the window
    pdftitle={My title},    % title
    pdfauthor={Author},     % author
    pdfsubject={Subject},   % subject of the document
    pdfcreator={Creator},   % creator of the document
    pdfproducer={Producer}, % producer of the document
    pdfkeywords={keyword1} {key2} {key3}, % list of keywords
    pdfnewwindow=true,      % links in new PDF window
    colorlinks=false,       % false: boxed links; true: colored links
    linkcolor=red,          % color of internal links (change box color with
 linkbordercolor)
    citecolor=green,        % color of links to bibliography
    filecolor=magenta,      % color of file links
    urlcolor=cyan           % color of external links
}
```

If you don't need such a high customization, here are some smaller but useful examples. When creating PDFs destined for printing, colored links are not a good thing as they end up in gray in the final output, making it difficult to read. You can use color frames, which are not printed:

```
\usepackage{hyperref}
\hypersetup{colorlinks=false}
```

or make links black:

```
\usepackage[hidelinks]{hyperref}
```

When you just want to provide information for the Document Info section of the PDF file, as well as enabling back references inside bibliography:

```
\usepackage[pdfauthor={Author's name},%
pdftitle={Document Title},%
pagebackref=true,%
pdftex]{hyperref}
```

By default, URLs are printed using mono-spaced fonts. If you don't like it and you want them to be printed with the same style of the rest of the text, you can use this:

```
\urlstyle{same}
```

20.4. Troubleshooting

20.4.1. Problems with Links and Equations 1

Messages like the following

```
! pdfTeX warning (ext4): destination with the same identifier (name{
equation.1.7.7.30}) has been already used, duplicate ignored
```

appear, when you have made something like

```
\begin{eqnarray}a=b\nonumber\end{eqnarray}
```

The error disappears, if you use instead this form:

```
\begin{eqnarray*}a=b\end{eqnarray*}
```

Beware that the shown line number is often completely different from the erroneous line.

Possible solution: Place the `amsmath` package before the `hyperref` package.

20.4.2. Problems with Links and Equations 2

Messages like the following

```
! Runaway argument?
{\@firstoffive }\fi ), Some text from your document here (\ref {re\ETC.
Latex Error: Paragraph ended before \Hy@setref@link was complete.
```

appear when you use `\label` inside an `align` environment.

Possible solution: Add the following to your preamble:

```
\AtBeginDocument{\let\textlabel\label}
```

20.4.3. Problems with Links and Pages

Messages like the following:

```
! pdfTeX warning (ext4): destination with the same
identifier (name{page.1}) has been already used,
duplicate ignored
```

appear when a counter gets reinitialized, for example by using the command `\mainmatter` provided by the book document class. It resets the page number counter to 1 prior to the first chapter of the book. But as the preface of the book also has a page number 1 all links to "page 1" would not be unique anymore, hence the notice that "duplicate has been ignored." The counter measure consists of putting `plainpages=false` into the `hyperref` options. This unfortunately only helps with the page counter. An even more radical solution is to use the option `hypertexnames=false`, but this will cause the page links in the index to stop working.

The best solution is to give each page a unique name by using the `\pagenumbering` command:

```
\pagenumbering{alph}    % a, b, c, ...
... titlepage, other front matter ...
\pagenumbering{roman}   % i, ii, iii, iv, ...
... table of contents, table of figures, ...
\pagenumbering{arabic}  % 1, 2, 3, 4, ...
... beginning of the main matter (chapter 1) ...
```

Another solution is to use `\pagenumbering{alph}` before the command `\maketitle`, which will give the title page the label page.a. Since the page number is suppressed, it won't make a difference to the output.

By changing the page numbering every time before the counter is reset, each page gets a unique name. In this case, the pages would be numbered a, b, c, i, ii, iii, iv, v, 1, 2, 3, 4, 5, ...

If you don't want the page numbers to be visible (for example, during the front matter part), use `\pagestyle{empty} ... \pagestyle{plain}`. The important point is that although the numbers are not visible, each page will have a unique name.

Another more flexible approach is to set the counter to something negative:

```
\setcounter{page}{-100}
... titlepage, other front matter ...
\pagenumbering{roman}   % i, ii, iii, iv, ...
... table of contents, table of figures, ...
\pagenumbering{arabic}  % 1, 2, 3, 4, ...
... beginning of the main matter (chapter 1) ...
```

which will give the first pages a unique negative number.

The problem can also occur with the `algorithms` package: because each algorithm uses the same line-numbering scheme, the line identifiers for the second and follow-on algorithms will be duplicates of the first.

The problem occurs with equation identifiers if you use `\nonumber` on every line of an eqnarray environment. In this case, use the *'ed form instead, e.g. `\begin{eqnarray*} ... \end{eqnarray*}` (which is an unnumbered equation array), and remove the now unnecessary `\nonumber` commands.

If your url's are too long and running off of the page, try using the `breakurl` package to split the url over multiple lines. This is especially important in a multicolumn environment where the line width is greatly shortened.

20.4.4. Problems with bookmarks

The text displayed by bookmarks does not always look like you expect it to look. Because bookmarks are "just text", much fewer characters are available for bookmarks than for normal LaTeX text. Hyperref will normally notice such problems and put up a warning:

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

You can now work around this problem by providing a text string for the bookmarks, which replaces the offending text:

```
\texorpdfstring{''TEX text''}{''Bookmark Text''}
```

Math expressions are a prime candidate for this kind of problem:

```
\section{ \texorpdfstring{$E=mc^2$}{E=mc2} }
```

which turns `\section{$E=mc^2$}` to `E=mc2` in the bookmark area. Color changes also do not travel well into bookmarks:

```
\section{ \textcolor{red}{Red !} }
```

produces the string ”redRed!”. The command `\textcolor` gets ignored but its argument (red) gets printed. If you use:

```
\section{ \texorpdfstring{\textcolor{red}{Red !}}{Red\ !} }
```

the result will be much more legible.

If you write your document in unicode and use the `unicode` option for the `hyperref` package you can use unicode characters in bookmarks. This will give you a much larger selection of characters to pick from when using `\texorpdfstring`.

20.4.5. Problems with tables and figures

The links created by `hyperref` point to the label created within the float environment, which, as previously described[8], must always be set after the caption. Since the caption is usually below a figure or table, the figure or table itself will not be visible upon clicking the link[9]. A workaround exists by using the package `hypcap` `http://www.ctan.org/tex-archive/macros/latex/contrib/oberdiek/hypcap.pdf` with:

```
\usepackage[all]{hypcap}
```

Be sure to call this package *after* loading `hyperref`.

If you use the `wrapfig` package[10] mentioned in the ”Wrapping text around figures[11]” section of the ”Floats, Figures and Captions” chapter, or other similar packages that define their own environments, you will need to manually include `\capstart` in those environments, *e.g.* :

```
\begin{wrapfigure}{R}{0.5\textwidth}
  \capstart
  \begin{center}
    \includegraphics[width=0.48\textwidth]{filename}
  \end{center}
  \caption{\label{labelname}a figure}
\end{wrapfigure}
```

8 Chapter 18.5 on page 238
9 `http://www.ctan.org/tex-archive/macros/latex/contrib/hyperref/README`
10 Wrapfig package webpage in CTAN ^{`http://www.ctan.org/tex-archive/macros/latex/contrib/wrapfig`}
11 Chapter 18.6 on page 239

20.4.6. Problems with long caption and \listoffigures or long title

There is an issue when using `\listoffigures` with `hyperref` for long captions or long titles. This happens when the captions (or the titles) are longer than the page width (about 7-9 words depending on your settings). To fix this, you need to use the option breaklinks when first declaring:

```
\usepackage[breaklinks]{hyperref}
```

This will then cause the links in the `\listoffigures` to word wrap properly.

20.4.7. Problems with already existing .toc, .lof and similar files

The format of some of the auxilliary files generated by latex changes when you include the `hyperref` package. One can therefore encounter errors like

```
! Argument of \Hy@setref@link has an extra }.
```

when the document is typeset with `hyperref` for the first time and these files already exist. The solution to the problem is to delete all the files that latex uses to get references right and typeset again.

20.4.8. Problems with footnotes and special characters

See the relevant section[12].

20.4.9. Problems with Beamer

Using the command

```
\hyperref[some_label]{some text}
```

is broken when pointed at a label. Instead of sending the user to the desired label, upon clicking the user will be sent to the first frame. A simple work around exists; instead of using

```
\phantomsection\label{some_label}
```

to label your frames, use

```
\hypertarget{some_label}{}
```

and reference it with

```
\hyperlink{some_label}{some text}
```

12 Chapter 19.1.3 on page 251

20.4.10. Problems with draft mode

WARNING! Please note that if you have activated the "draft"-option in your \document-class declaration the hyperlinks will not show up in the table of contents!!!

20.5. Notes and References

21. Labels and Cross-referencing

21.1. Introduction

In LaTeX you can easily reference almost anything that is numbered (sections, figures, formulas), and LaTeX will take care of numbering, updating it whenever necessary. The commands to be used do not depend on what you are referencing, and they are:

`\label{`*`marker`*` }`

you give the object you want to reference a *marker* , you can see it like a name.

`\ref{`*`marker`*` }`

you can reference the object you have *marked* before. This prints the number that was assigned to the object.

`\pageref{`*`marker`*` }`

It will print the number of the page where the object is.

LaTeX will calculate the right numbering for the objects in the document; the *marker* you have used to label the object will not be shown anywhere in the document. Then LaTeX will replace the string "`\ref{`*`marker`*` }` " with the right number that was assigned to the object. If you reference a *marker* that does not exist, the compilation of the document will be successful but LaTeX will return a warning:

```
LaTeX Warning: There were undefined references.
```

and it will replace "`\ref{`*`unknown-marker`*` }` " with "??" (so it will be easy to find in the document).

As you may have noticed reading how it works, it is a two-step process: first the compiler has to store the labels with the right number to be used for referencing, then it has to replace the `\ref` with the right number. That is why, when you use references, you have to compile your document twice to see the proper output. If you compile it only once, LaTeX will use the older information it collected in previous compilations (that might be outdated), but the compiler will inform you printing on the screen at the end of the compilation:

```
LaTeX Warning: Label(s) may have changed. Rerun to get cross-references
right.
```

Using the command `\pageref{}` you can help the reader to find the referenced object by providing also the page number where it can be found. You could write something like:

```
See figure~\ref{fig:test} on page~\pageref{fig:test}.
```

Since you can use exactly the same commands to reference almost anything, you might get a bit confused after you have introduced a lot of references. It is common practice among LaTeX users to add a few letters to the label to describe *what* you are referencing. Some packages, such as `fancyref` , rely on this meta information. Here is an example:

`ch:`	chapter
`sec:`	section
`subsec:`	subsection
`fig:`	figure
`tab:`	table
`eq:`	equation
`lst:`	code listing
`itm:`	enumerated list item
`alg:`	algorithm
`app:`	appendix subsection

Following this convention, the label of a figure will look like `\label{fig:`*`my_figure`*` }` , etc. You are not obligated to use these prefixes. You can use any string as argument of `\label{...}` , but these prefixes become increasingly useful as your document grows in size.

Another suggestion: try to avoid using numbers within labels. You are better off describing *what* the object is about. This way, if you change the order of the objects, you will not have to rename all your labels and their references.

If you want to be able to see the markers you are using in the output document as well, you can use the `showkeys` package; this can be very useful while developing your document. For more information see the Packages[1] section.

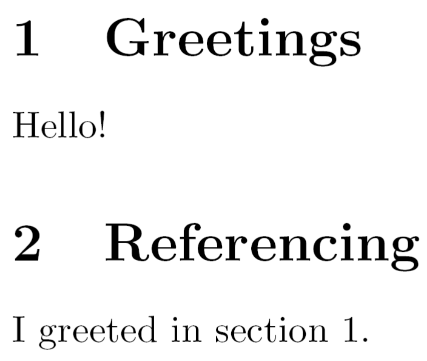
21.2. Examples

Here are some practical examples, but you will notice that they are all the same because they all use the same commands.

21.2.1. Sections

```
\section{Greetings}
\label{sec:greetings}

Hello!

\section{Referencing}

I greeted in section~\ref{sec:greetings}.
```

1 http://en.wikibooks.org/wiki/LaTeX%2FPackages


1 Greetings

Hello!

2 Referencing

I greeted in section 1.

Figure 75

You could place the label anywhere in the section; however, in order to avoid confusion, it is better to place it immediately after the beginning of the section. Note how the marker starts with *sec:* , as suggested before. The label is then referenced in a different section. The tilde (˜) indicates a non-breaking space[2].

21.2.2. Pictures

You can reference a picture by inserting it in the `figure` floating environment.

```
\begin{figure}
  \centering
    \includegraphics[width=0.5\textwidth]{gull}
  \caption{Close-up of a gull}
  \label{fig:gull}
\end{figure}
Figure \ref{fig:gull} shows a photograph of a gull.
```

2 http://en.wikipedia.org/wiki/non-breaking%20space

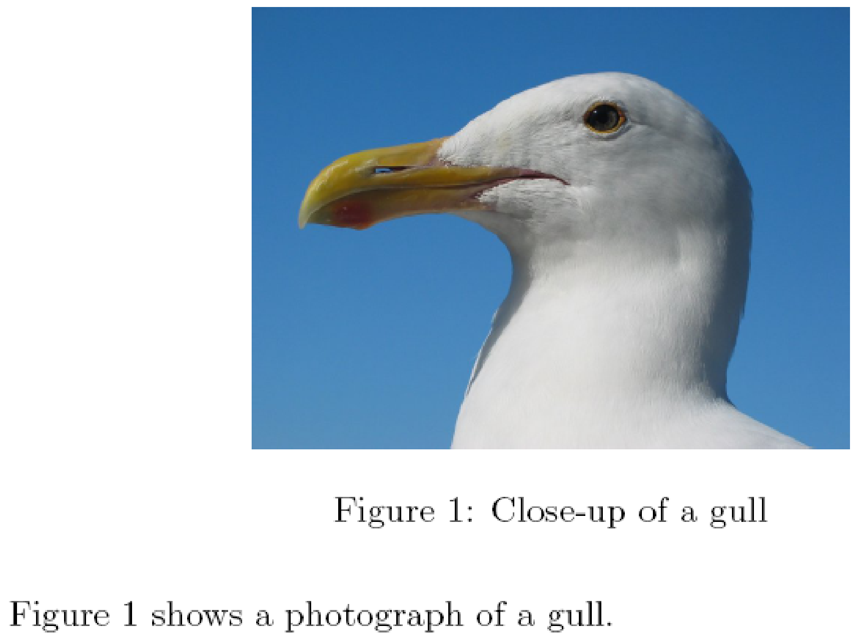


Figure 1: Close-up of a gull

Figure 1 shows a photograph of a gull.

Figure 76

When a label is declared within a float environment, the `\ref{...}` will return the respective fig/table number, but it must occur **after** the caption. When declared outside, it will give the section number. To be completely safe, the label for any picture or table can go within the `\caption{}` command, as in

```
\caption{Close-up of a gull\label{fig:gull}}
```

See the Floats, Figures and Captions[3] section for more about the `figure` and related environments.

Fixing wrong labels

The command `\label` must appear after (or inside) `\caption` . Otherwise, it will pick up the current section or list number instead of what you intended.

```
\begin{figure}
  \centering
  \includegraphics[width=0.5\textwidth]{gull}
  \caption{Close-up of a gull} \label{fig:gull}
\end{figure}
```

Issues with links to tables and figures handled by hyperref

In case you use the package `hyperref` to create a PDF, the links to tables or figures will point to the caption of the table or figure, which is always below the table or figure itself[4].

3 Chapter 18 on page 231
4 `http://www.ctan.org/tex-archive/macros/latex/contrib/hyperref/README`

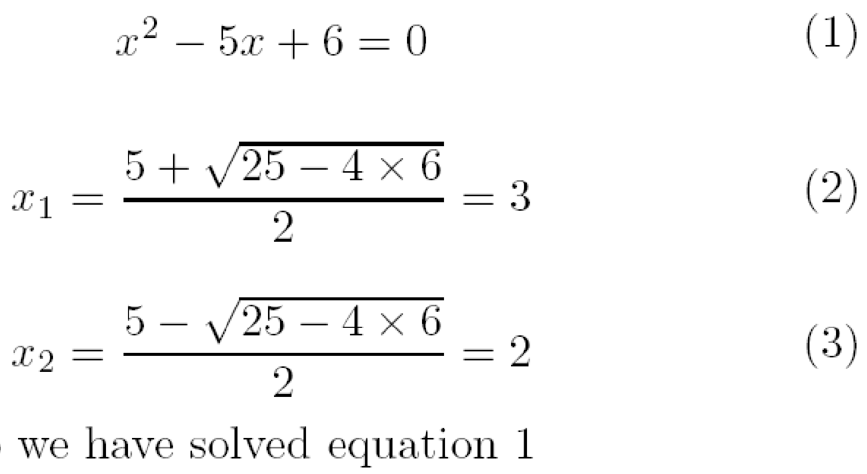
Therefore the table or figure will not be visible, if it is above the pointer and one has to scroll up in order to see it. If you want the link point to the top of the image you can give the option `hypcap` to the `caption` package:

```
\usepackage[hypcap]{caption}
```

21.2.3. Formulae

Here is an example showing how to reference formulae:

```
\begin{equation} \label{eq:solve}
x^2 - 5 x + 6 = 0
\end{equation}

\begin{equation}
x_1 = \frac{5 + \sqrt{25 - 4 \times 6}}{2} = 3
\end{equation}

\begin{equation}
x_2 = \frac{5 - \sqrt{25 - 4 \times 6}}{2} = 2
\end{equation}

and so we have solved equation~\ref{eq:solve}
```



$$x^2 - 5x + 6 = 0 \qquad (1)$$

$$x_1 = \frac{5 + \sqrt{25 - 4 \times 6}}{2} = 3 \qquad (2)$$

$$x_2 = \frac{5 - \sqrt{25 - 4 \times 6}}{2} = 2 \qquad (3)$$

and so we have solved equation 1

Figure 77

As you can see, the label is placed soon after the beginning of the math mode. In order to reference a formula, you have to use an environment that adds numbers. Most of the times you will be using the `equation` environment; that is the best choice for one-line formulae, whether you are using `amsmath` or not. Note also the *eq:* prefix in the label.

`eqref`

The `amsmath` package adds a new command for referencing formulae; it is `\eqref{}` . It works exactly like `\ref{}` , but it adds parentheses so that, instead of printing a plain number as *5* , it will print *(5)* . This can be useful to help the reader distinguish between formulae and other things, without the need to repeat the word "formula" before any reference. Its output can be changed as desired; for more information see the `amsmath` documentation.

tag

The `\tag{eqnno}` command is used to manually set equation numbers where *eqnno* is the arbitrary text string you want to appear in the document. It is normally better to use labels, but sometimes hard-coded equation numbers might offer a useful work-around. This may for instance be useful if you want to repeat an equation that is used before, e.g. `\tag{\ref{eqn:before}}` .

numberwithin

The `amsmath` package adds the `\numberwithin{countera}{counterb}` command which replaces the simple `countera` by a more sophisticated `counterb.countera` . For example `\numberwithin{equation}{section}` in the preamble will prepend the section number to all equation numbers.

cases

The `cases` package adds the `\numcases` and the `\subnumcases` commands, which produce multi-case equations with a separate equation number and a separate equation number plus a letter, respectively, for each case.

21.3. The `varioref` package

The `varioref` package introduces a new command called `\vref{}` . This command is used exactly like the basic `\ref` , but it has a different output according to the context. If the object to be referenced is in the same page, it works just like `\ref` ; if the object is far away it will print something like "5 on page 25", i.e. it adds the page number automatically. If the object is close, it can use more refined sentences like "on the next page" or "on the facing page" automatically, according to the context and the document class.

This command has to be used very carefully. It outputs more than one word, so it may happen its output falls on two different pages. In this case, the algorithm can get confused and cause a loop. Let's make an example. You label an object on page 23 and the `\vref` output happens to stay between page 23 and 24. If it were on page 23, it would print like the basic `ref` , if it were on page 24, it would print "on the previous page", but it is on both, and this may cause some strange errors at compiling time that are very hard to be fixed. You could think that this happens very rarely; unfortunately, if you write a long document it is not uncommon to have hundreds of references, so situations like these are likely to happen. One way to avoid problems during development is to use the standard `ref` all the time, and convert it to `vref` when the document is close to its final version, and then making adjustments to fix possible problems.

21.4. The `hyperref` package

21.4.1. `autoref`

The `hyperref`[5] package introduces another useful command; `\autoref{}` . This command creates a reference with additional text corresponding to the target's type, all of which will be a hyperlink. For example, the command `\autoref{sec:intro}` would create a hyperlink to the `\label{sec:intro}` command, wherever it is. Assuming that this label is pointing to a section, the hyperlink would contain the text "section 3.4", or similar (the full list of default names can be found here[6]). Note that, while there's an

```
\autoref*
```

command that produces an unlinked prefix (useful if the label is on the same page as the reference), no alternative

```
\Autoref
```

command is defined to produce capitalized versions (useful, for instance, when starting sentences); but since the capitalization of autoref names was chosen by the package author, you can customize the prefixed text by redefining `\`*`type`* `autorefname` to the prefix you want, as in:

```
\def\sectionautorefname{Section}
```

This renaming trick can, of course, be used for other purposes as well.

- If you would like a hyperlink reference, but do not want the predefined text that `\autoref{}` provides, you can do this with a command such as `\hyperref[sec:intro]{Appendix~\ref*{sec:intro}}` . Note that you can disable the creation of hyperlinks in `hyperref` , and just use these commands for automatic text.
- Keep in mind that the \label **must** be placed inside an environment with a counter, such as a table or a figure. Otherwise, not only the number will refer to the current section, as mentioned above[7], but the name will refer to the previous environment with a counter. For example, if you put a label after closing a figure, the label will still say "figure n", on which n is the current section number.

21.4.2. `nameref`

The `hyperref` package also automatically includes the `nameref` package, and a similarly named command. It is similar to `\autoref{}` , but inserts text corresponding to the section name, for example.

5 `http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FHyperref`
6 `http://www.tug.org/applications/hyperref/manual.html#TBL-24`
7 Chapter 21.2.2 on page 270

Input:

```
\section{MyFirstSection} \label{sec:marker}
\section{MySecondSection}
In section~\nameref{sec:marker} we defined...
```

Output:

In section MyFirstSection we defined...

21.4.3. Anchor manual positioning

When you define a `\label` outside a figure, a table, or other floating objects, the label points to the current section. In some cases, this behavior is not what you'd like and you'd prefer the generated link to point to the line where the `\label` is defined. This can be achieved with the command `\phantomsection` as in this example:

```
%The link location will be placed on the line below.
\phantomsection
\label{the_label}
```

21.5. The `cleveref` package

The `cleveref` package introduces the new command `\cref{}` which includes the type of referenced object like `\autoref{}` does. The alternate `\labelcref{}` command works more like standard `\ref{}` . References to pages are handled by the `\cpageref{}` command.

The `\crefrange{}{}` and `\cpagerefrange{}` commands expect a start and end label in either order and provide a natural language (`babel` enabled) range. If labels are enumerated as a comma-separated list with the usual `\cref{}` command, it will sort them and group into ranges automatically.

The format can be specified in the preamble.

21.6. See also

- LaTeX/Glossary[8]

21.7. Notes and References

8 Chapter 35 on page 429

Part III.

Mechanics

22. Errors and Warnings

LaTeX describes what it is typesetting while it does it. If it encounters something it doesn't understand or can't do, it will display a message saying what is wrong. It may also display warnings for less serious conditions.

Don't panic if you see error messages : it is very common to mistype or misspell commands, forget curly braces, type a forward slash instead of a backslash, or use a special character by mistake. Errors are easily spotted and easily corrected in your editor, and you can then run LaTeX again to check you have fixed everything. Some of the most common errors are described in next sections.

22.1. Error messages

The format of an error message is always the same. Error messages begin with an exclamation mark at the start of the line, and give a description of the error, followed by another line starting with the number, which refers to the line-number in your document file which LaTeX was processing when the error was spotted. Here's an example, showing that the user mistyped the `\tableofcontents` command:

```
! Undefined control sequence.
l.6 \tableofcotnetns
```

When LaTeX finds an error like this, it displays the error message and pauses. You must type one of the following letters to continue:

Key	**Meaning**
x	Stop immediately and e**x** it the program.
q	Carry on **q** uietly as best you can and don't bother me with any more error messages.
e	Stop the program but re-position the text in my **e** ditor at the point where you found the error (This only works if you're using an editor which LaTeX can communicate with).
h	Try to give me more **h** elp.
i	(followed by a correction) means **i** nput the correction in place of the error and carry on (This is only a temporary fix to get the file processed. You still have to make that correction in the editor).
r	**r** un in non-stop mode. Plow through any errors, unless too many pile up and it fails (100 errors).

Some systems (Emacs is one example) run LaTeX with a ”nonstop” switch turned on, so it will always process through to the end of the file, regardless of errors, or until a limit is reached.

22.2. Warnings

Warnings don’t begin with an exclamation mark: they are just comments by LaTeX about things you might want to look into, such as overlong or underrun lines (often caused by unusual hyphenations, for example), pages running short or long, and other typographical niceties (most of which you can ignore until later). Unlike other systems, which try to hide unevennesses in the text (usually unsuccessfully) by interfering with the letter spacing, LaTeX takes the view that the author or editor should be able to contribute. While it is certainly possible to set LaTeX’s parameters so that the spacing is sufficiently sloppy that you will almost never get a warning about badly-fitting lines or pages, you will almost certainly just be delaying matters until you start to get complaints from your readers or publishers.

22.3. Examples

Only a few common error messages are given here: those most likely to be encountered by beginners. If you find another error message not shown here, and it’s not clear what you should do, ask for help.

Most error messages are self-explanatory, but be aware that the place where LaTeX spots and reports an error may be later in the file than the place where it actually occurred. For example if you forget to close a curly brace which encloses, say, italics, LaTeX won’t report this until something else occurs which can’t happen until the curly brace is encountered (e.g. the end of the document!) Some errors can only be righted by humans who can read and understand what the document is supposed to mean or look like.

Newcomers should remember to check the list of special characters: a very large number of errors when you are learning LaTeX are due to accidentally typing a special character when you didn’t mean to. This disappears after a few days as you get used to them.

22.3.1. Too many }’s

```
! Too many }'s.
l.6 \date December 2004}
```

The reason LaTeX thinks there are too many }’s here is that the opening curly brace is missing after the `\date` control sequence and before the word December, so the closing curly brace is seen as one too many (which it is!). In fact, there are other things which can follow the `\date` command apart from a date in curly braces, so LaTeX cannot possibly guess that you’ve missed out the opening curly brace until it finds a closing one!

22.3.2. Undefined control sequence

```
! Undefined control sequence.
l.6 \dtae
{December 2004}
```

In this example, LaTeX is complaining that it has no such command (”control sequence”) as `\dtae` . Obviously it’s been mistyped, but only a human can detect that fact: all LaTeX knows is that `\dtae` is not a command it knows about: it’s undefined. Mistypings are the most common source of errors. Some editors allow common commands and environments to be inserted using drop-down menus or icons, which may be used to avoid these errors.

22.3.3. Not in Mathematics Mode

```
! Missing $ inserted
```

A character that can only be used in the mathematics was inserted in normal text. If you intended to use mathematics mode, then use `$...$` or `\begin{math}...\end{math}` or use the ’quick math mode’: `\ensuremath{...}` . If you did not intend to use mathematics mode, then perhaps you are trying to use a special character[1] that needs to be entered in a different way; for example `_` will be interpreted as a subscript operator in mathematics mode, and you need `\_` to get an underscore character.

This can also happen if you use the wrong character encoding, for example using utf8 without ”\usepackage[utf8]{inputenc}” or using iso8859-1 without ”\usepackage[latin1]{inputenc}”, there are several character encoding formats, make sure to pick the right one.

22.3.4. Runaway argument

```
Runaway argument?
{December 2004 \maketitle
! Paragraph ended before \date was complete.
<to be read again>
\par
l.8
```

In this error, the closing curly brace has been omitted from the date. It’s the opposite of the error of too many }’s, and it results in `\maketitle` trying to format the title page while LaTeX is still expecting more text for the date! As \maketitle creates new paragraphs on the title page, this is detected and LaTeX complains that the previous paragraph has ended but \date is not yet finished.

1 Chapter 4.5.6 on page 48

22.3.5. Underfull hbox

```
Underfull \hbox (badness 1394) in paragraph
at lines 28--30
[][]\LY1/brm/b/n/10 Bull, RJ: \LY1/brm/m/n/10
Ac-count-ing in Busi-
[94]
```

This is a warning that LaTeX cannot stretch the line wide enough to fit, without making the spacing bigger than its currently permitted maximum. The badness (0-10,000) indicates how severe this is (here you can probably ignore a badness of 1394). It says what lines of your file it was typesetting when it found this, and the number in square brackets is the number of the page onto which the offending line was printed. The codes separated by slashes are the typeface and font style and size used in the line. Ignore them for the moment.

This comes up if you force a linebreak, e.g., \\, and have a return before it. Normally TeX ignores linebreaks, providing full paragraphs to ragged text. In this case it is necessary to pull the linebreak up one line to the end of the previous sentence.

This warning may also appear when inserting images. It can be avoided by using the \textwidth or possibly \linewidth options, e.g. \includegraphics[width=\textwidth]{image_name}

22.3.6. Overfull hbox

```
[101]
Overfull \hbox (9.11617pt too wide) in paragraph
at lines 860--861
[]\LY1/brm/m/n/10 Windows, \LY1/brm/m/it/10 see
\LY1/brm/m/n/10 X Win-
```

An overfull \hbox means that there is a hyphenation or justification problem: moving the last word on the line to the next line would make the spaces in the line wider than the current limit; keeping the word on the line would make the spaces smaller than the current limit, so the word is left on the line, but with the minimum allowed space between words, and which makes the line go over the edge.

The warning is given so that you can find the line in the code that originates the problem (in this case: 860-861) and fix it. The line on this example is too long by a shade over 9pt. The chosen hyphenation point which minimizes the error is shown at the end of the line (Win-). Line numbers and page numbers are given as before. In this case, 9pt is too much to ignore (over 3mm), and a manual correction needs making (such as a change to the hyphenation), or the flexibility settings need changing.

If the ”overfull” word includes a forward slash, such as ”`input/output` ”, this should be properly typeset as ”`input\slash output` ”. The use of `\slash` has the same effect as using the ”`/` ” character, except that it can form the end of a line (with the following words appearing at the start of the next line). The ”`/` ” character is typically used in units, such as ”`mm/year` ” character, which should not be broken over multiple lines.

The warning can also be issued when the \end{document} tag was not included or was deleted.

Easily spotting overfull hboxes in the document

To easily find the location of overfull hbox in your document, you can make latex add a black bar where a line is too wide:

```
\overfullrule=2cm
```

22.3.7. Missing package

```
! LaTeX Error: File `paralisy.sty' not found.
Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: sty)
Enter file name:
```

When you use the `\usepackage` command to request LaTeX to use a certain package, it will look for a file with the specified name and the filetype `.sty` . In this case the user has mistyped the name of the paralist package, so it's easy to fix. However, if you get the name right, but the package is not installed on your machine, you will need to download and install it before continuing. If you don't want to affect the global installation of the machine, you can simply download from Internet the necessary `.sty` file and put it in the same folder of the document you are compiling.

22.3.8. Package babel Warning: No hyphenation patterns were loaded for the language X

Although this is a warning from the Babel package and not from LaTeX, this error is very common and (can) give some strange hyphenation (word breaking) problems in your document. Wrong hyphenation rules can decrease the neatness of your document.

```
Package babel Warning: No hyphenation patterns were loaded for
(babel)                the language `Latin'
(babel)                I will use the patterns loaded for \language=0 instead.
```

This can happen after the usage of: (see LaTeX/Internationalization[2])

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

The solution is not difficult, just install the used language in your LaTeX distribution[3].

2 Chapter 12 on page 133
3 Chapter 2.3.1 on page 18

22.3.9. Package babel Error: You haven't loaded the option X yet.

If you previously set the X language, and then decided to switch to Y, you will get this error. This may seem awkward, as there is obviously no error in your code if you did not change anything. The answer lies in the `.aux` file, where babel defined your language. If you try the compilation a second time, it should work. If not, delete the `.aux` file, then everything will work as usual.

22.3.10. No error message, but won't compile

One common cause of (pdf)LaTeX getting stuck is forgetting to include

```
\end{document}
```

22.4. Software that can check your .tex Code

There are several programs capable of checking LaTeX source, with the aim of finding errors or highlighting bad practice, and providing more help to (particularly novice) users than the built-in error messages.

- nag (www.ctan.org/tex-archive/macros/latex/contrib/nag[4]) is a LaTeX package designed to indicate the use of obsolete commands.
- lacheck (www.ctan.org/tex-archive/support/lacheck[5]) is a consistency checker intended to spot mistakes in code. It is available as source code or compiled for Windows and OS/2
- chktex (baruch.ev-en.org/proj/chktex/[6]) is a LaTeX semantic checker available as source code for Unix-like systems.

4 `http://www.ctan.org/tex-archive/macros/latex/contrib/nag`
5 `http://www.ctan.org/tex-archive/support/lacheck`
6 `http://baruch.ev-en.org/proj/chktex/`

23. Lengths

In TeX, a length is

- a floating point number followed by a unit, optionally followed by a stretching value;

```
3.5pt plus 1pt minus 2pt
```

- a floating point factor followed by a macro that expands to a length.

```
1.7\textwidth
```

23.1. Units

First, we introduce the LaTeX measurement units. All LaTeX units are two-letter abbreviations. You can choose from a variety of units. Here are the most common ones.[1]

Abbreviation	Definition	Value in points (pt)
pt	a point is 1/72.27 inch, that means about 0.0138 inch or 0.3515 mm. 1pt is the default length.	1
mm	a millimeter	2.84
cm	a centimeter	28.4
in	inch	72.27
ex	roughly the height of an 'x' **in the current font**	*undefined, depends on the font used*
em	roughly the width of an 'M' (uppercase) **in the current font**	*undefined, depends on the font used*

And here are some less common units.[2]

Abbreviation	Definition	Value in points (pt)
bp	a big point is 1/72 inch, that means about 0.0139 inch or 0.3527 mm.	1.00375

1 http://www.giss.nasa.gov/tools/latex/ltx-86.html
2 http://www.giss.nasa.gov/tools/latex/ltx-86.html

Abbreviation	Definition	Value in points (pt)
pc	pica	12
dd	didôt (1157 didôt = 1238 points)	1.07
cc	cîcero (12 didôt)	12.84
sp	scaled point (65536sp per point)	0.000015

23.2. Box lengths

A box in TeX is characterized by three lengths:

- *depth*
- *height*
- *width*

See Boxes[3].

23.3. Length manipulation

You can change the values of the variables defining the page layout with two commands. With this one you can set a new value for an existing length variable:

```
\setlength{\mylength}{length}
```

with this other one, you can add a value to the existing one:

```
\addtolength{\mylength}{length}
```

You can create your own length with the command, and you must create a new length before you attempt to set it:

```
\newlength{\mylength}
```

You may also set a length from the size of a text with one of these commands:

```
\settowidth{\mylength}{some text}
\settoheight{\mylength}{some text}
\settodepth{\mylength}{some text}
```

When using these commands, you may duplicate the text that you want to use as reference if you plan to also display it. But LaTeX also provides `\savebox` to avoid this duplication. You may wish to look at the example below to see how you can use these. See Boxes[4] for more details.

You can also define stretched values. A stretching value is a length preceded by `plus` or `minus` to specify to what extent `tex` is authorized to change the length. Example:

3 Chapter 25 on page 295
4 Chapter 25 on page 295

```
\setlength{\parskip}{10pt plus 5pt minus 3pt}
```

It means that `tex` will try to use a length of 10pt; if it is underfull, it will raise the length up to a maximum of 15pt; if it is overfull, it will lower the length up to a minimum of 7pt.

Note that it is not mandatory to specify both the `plus` and the `minus` values, but if you do, `latxpar` must be placed before `minus`.

To print a length, you can use the `\the` command:

```
\the\textwidth
```

23.3.1. Plain TeX

To create a new length:

```
\newdimen\mylength
```

To set a length:

```
\mylength=1.5in
```

To view, it is the same as with LaTeX, using the command `\the`.

23.4. LaTeX default lengths

Common length macros are:

\baselineskip

The normal vertical distance between lines in a paragraph.

\baselinestretch

Multiplies \baselineskip.

\columnsep

The distance between columns.

\columnwidth

The width of the column.

\evensidemargin

The margin for 'even' pages (think of a printed booklet).

\linewidth

The width of a line in the local environment.

\oddsidemargin

The margin for 'odd' pages (think of a printed booklet).

\paperwidth

The width of the page.

\paperheight

The height of the page.

\parindent

The normal paragraph indentation.

\parskip

The extra vertical space between paragraphs.

\tabcolsep

The default separation between columns in a tabular environment.

\textheight

The height of text on the page.

\textwidth

The width of the text on the page.

\topmargin

The size of the top margin.

\unitlength

Units of length in `picture` environment.

23.5. Fixed-length spaces

To insert a fixed-length space, use:

```
\hspace{length}
\vspace{length}
```

`\hspace` stands for horizontal space, `\vspace` for vertical space.

If such a space should be kept even if it falls at the end or the start of a line, use `\hspace*` instead.

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`. If you want to add space at the beginning of the document, without anything else written before, then you may use

```
{ \vspace*{length} }
```

It's important you use the `\vspace*` command instead of `\vspace`, otherwise LaTeX can silently ignore the extra space.

TeX features some macros for fixed-length spacing.

`\smallskip`

Inserts a small space in vertical mode (between two paragraphs).

`\medskip`

Inserts a medium space in vertical mode (between two paragraphs).

`\bigskip`

Inserts a big space in vertical mode (between two paragraphs).

The vertical mode is during the process of assembling boxes ”vertically”, like paragraphs to build a page. The horizontal mode is during the process of assembling boxes ”horizontally”, like letters to build a word or words to build a paragraph.

The fact they are vertical mode commands mean they will be ignored (or fail) in horizontal mode such as in the middle of a paragraph. The first token next the a double linebreak is still in vertical mode if it does not expand to characters.

```
% WRONG!
Some words.
\bigskip
Let's continue.

%% CORRECT!
Some words.

\bigskip
Let's continue.
```

⚠ **Warning**

This is a common error! Anyway, these commands should never be used in regular documents.

23.6. Rubber/Stretching lengths

The command:

```
\stretch{factor}
```

generates a special rubber space where `factor` is a number, possibly a float. It stretches until all the remaining space on a line is filled up. If two `\hspace{\stretch{factor}}` 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
```

The same way, you can stretch vertically:

```
\maketitle
\vspace{ \stretch{1} }
Some comments.
\vspace{ \stretch{1} }
\tableofcontents
```

You can also use `\fill` instead of `\stretch{1}`.

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

There are 'shortcut commands' for stretching with factor 1 (*i.e.* with `\stretch{1}` or `\fill`): `\hfill` and `\vfill`.

Example:

```
\maketitle
\vfill
Some comments.
\vfill
\tableofcontents
```

23.6.1. Fill the rest of the line

Several macros allow filling the rest of the line -- or stretching parts of the line -- in different manners.

- `\hfill` will produce empty space.
- `\dotfill` will produce dots.
- `\hrulefill` will produce a rule.

23.7. Examples

Resize an image to take exactly half the text width :

```
\includegraphics[width=0.5\textwidth]{mygraphic}
```

Make distance between items larger (inside an itemize environment) :

```
\addtolength{\itemsep}{0.5\baselineskip}
```

Use of `\savebox` to resize an image to the height of the text:

```
% Create the holders we will need for our work
\newlength{\mytitleheight}
\newsavebox{\mytitletext}
% Create the reference text for measures
\savebox{\mytitletext}{%
  \Large\bfseries This is our title%
}
\settoheight{\mytitleheight}{ \usebox{\mytitletext} }
% Now creates the actual object in our document
\framebox[\textwidth][l]{%
  \includegraphics[height=\mytitleheight]{my_image}%
  \hspace{2mm}%
  \usebox{\mytitletext}%
}
```

23.8. References

23.9. See also

- University of Cambridge > Engineering Department > computing help > LaTeX > Squeezing Space in LaTeX[5]

5 `http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/squeeze.html`

24. Counters

Counters are an essential part of LaTeX: they allow you to control the numbering mechanism of everything (sections, lists, captions, etc.).

24.1. Counter manipulation

In LaTeX it is fairly easy to create new counters and even counters that reset automatically when another counter is increased (think subsection in a section for example). With the command

```
\newcounter{NameOfTheNewCounter}
```

you create a new counter that is automatically set to zero. If you want the counter to be reset to zero every time another counter is increased, use:

```
\newcounter{NameOfTheNewCounter}[NameOfTheOtherCounter]
```

To increase the counter, either use

```
\stepcounter{NameOfTheNewCounter}
```

or

```
\refstepcounter{NameOfTheNewCounter} % used for labels and cross referencing
```

or

```
\addtocounter{NameOfTheNewCounter}{number}
```

here the number can also be negative. For automatic resetting you need to use `\stepcounter` .

To set the counter value explicitly, use

```
\setcounter{NameOfTheNewCounter}{number}
```

24.2. Counter access

There are several ways to get access to a counter.

- `\theNameOfTheNewCounter` will print the formatted string related to the counter (note the "the" before the actual name of the counter).
- `\value{NameOfTheNewCounter}` will return the counter value which can be used by other counters or for calculations. It is not a formatted string, so it cannot be used in text.

- `\arabic{NameOfTheNewCounter}` will print the formatted counter using arabic numbers.

Note that `\arabic{NameOfTheNewCounter}` may be used as a value too, but not the others.

Strangely enough, LaTeX counters are *not* introduced by a backslash in any case, even with the `\the` command. plainTeX equivalents `\count` and `\newcounter\mycounter` do abide by the backslash rule.

24.3. Counter style

Each counter also has a default format that dictates how it is displayed whenever LaTeX needs to print it. Such formats are specified using internal LaTeX commands:

Command	**Example**
`\arabic`	1, 2, 3 ...
`\alph`	a, b, c ...
`\Alph`	A, B, C ...
`\roman`	i, ii, iii ...
`\Roman`	I, II, III ...
`\fnsymbol`	Aimed at footnotes; prints a sequence of symbols.

24.4. LaTeX default counters

- part
- chapter
- section
- subsection
- subsubsection
- paragraph
- subparagraph
- page
- equation
- figure
- table
- footnote
- mpfootnote

For the `enumerate` environment:

- enumi
- enumii
- enumiii
- enumiv

24.5. Book with parts, sections, but no chapters

Here follows an example where we want to use parts and sections, but no chapters in the book class :

```
\renewcommand{\thesection}{\thepart .\arabic{section}}

\part{My Part}
\section{My Section}
\subsection{My Subsection}
```

24.6. Custom *enumerate*

See the List Structures[1] chapter.

24.7. Custom sectioning

Here is an example for recreating something similar to a section and subsection counter that already exist in LaTeX:

```
\newcounter{mysection}
\newcounter{mysubsection}[mysection]
\addtocounter{mysection}{2} % set them to some other numbers than 0
\addtocounter{mysubsection}{10} % same
%
\arabic{mysection}.\arabic{mysubsection}
Blah blah

\stepcounter{mysection}
\arabic{mysection}.\arabic{mysubsection}
Blah blah

\stepcounter{mysubsection}
\arabic{mysection}.\arabic{mysubsection}
Blah blah

\addtocounter{mysubsection}{25}
\arabic{mysection}.\arabic{mysubsection}
Blah blah and more blah blah
```

1 Chapter 10 on page 109

25. 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.

Admittedly, this is a very simplistic description of what really happens, but the point is that TeX operates with glue and boxes. Letters are not the only things that can be boxes. One can put virtually everything into a box, including other boxes. Each box will then be handled by LaTeX as if it were a single letter.

The past chapters have already dealt with some boxes, although they weren't described as such. The tabular environment and the `\includegraphics`, for example, both produce a box. This means that one 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`.

25.1. TeX character boxes

TeX characters are stored in boxes like every printed element. Boxes have three dimensional properties:

- The *height* is the length between the baseline and the top of the box.
- The *depth* is the length between the baseline and the bottom of the box.
- The *width* is the width of the box.

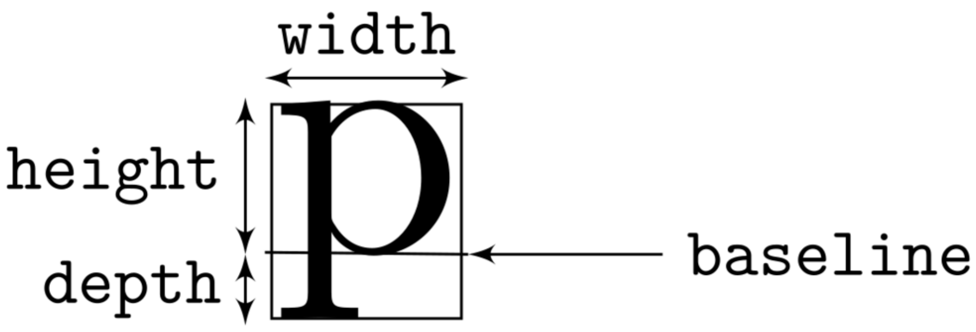


Figure 78

25.2. makebox and mbox

While `\parbox` packs up a whole paragraph doing line breaking and everything, there is also a class of boxing commands that 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. (See Hyphenation[1].) As you can put boxes inside boxes, these horizontal box packers give you ultimate flexibility.

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

width defines the width of the resulting box as seen from the outside. 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. Besides the length[2] 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, flush**l** eft, flush**r** ight, or **s** pread the text to fill the box.

```
\makebox[0pt]{Some text} over this text

\makebox[15ex][s]{Censored text}\hspace{-15ex}\makebox[15ex][s]{X X X X X}

Text \makebox[2\width][r]{running away}
```

25.3. framebox

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

```
\fbox{text}
\framebox[width][pos]{text}
```

The following example shows you some things you could do with the `\makebox` and `\frame-box` commands:

```
 \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?
```

1 Chapter 6.2 on page 67
2 Chapter 23 on page 283

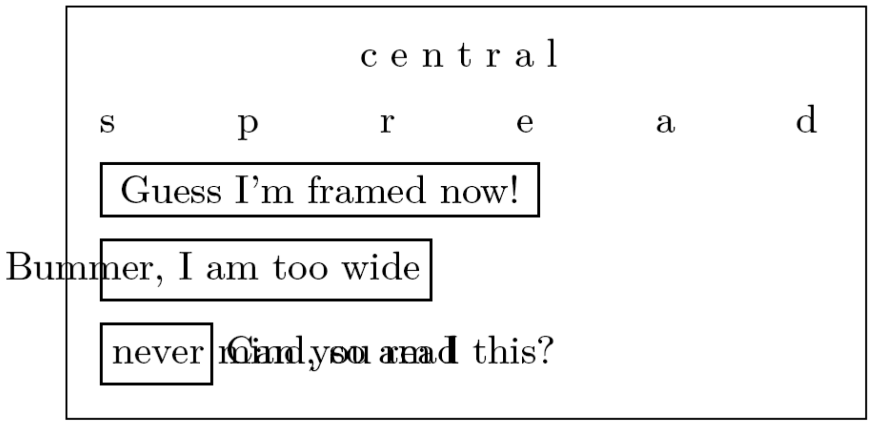


Figure 79

You can tweak the following frame lengths.

- `\fboxsep`: the distance between the frame and the content.
- `\fboxrule`: the thickness of the rule.

This prints a thick and more distant frame:

```
\setlength{\fboxsep}{10pt}
\setlength{\fboxrule}{5pt}
\fbox{A frame.}
```

This shows the box frame of a letter.

```
\setlength{\fboxsep}{0pt}
\fbox{A}
```

25.4. framed

An alternative to these approaches is the usage of the `framed` environment (you will need to include the `framed` package to use it). This provides an easy way to box a paragraph within a document:

```
\usepackage{framed}
% ...

\begin{framed}
This is an easy way to box text within a document!
\end{framed}
```

You can do it manually with a parbox[3].

3 Chapter 25.6 on page 298

25.5. raisebox

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

```
\raisebox{lift}[height][depth]{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. The two optional parameters set for the height and depth of the raisebox. For instance you can observe the difference when embedded in a framebox.

```
\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.
```

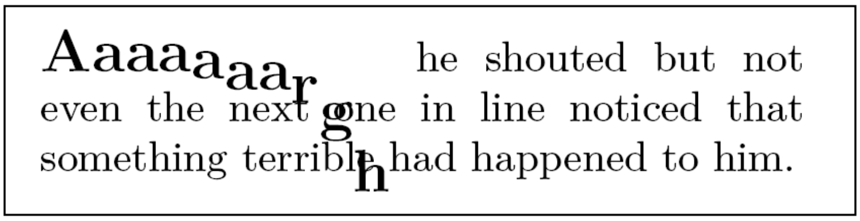

Aaaaaaargh he shouted but not even the next one in line noticed that something terrible had happened to him.

Figure 80

25.6. minipage and parbox

Most standard LaTeX boxes are not *long* commands, *i.e.* they do not support breaks nor paragraphs. However you can pack a paragraph of your choice into a box with either the `\parbox[pos][height][contentpos]{width}{text}` command or the `\begin{minipage}[pos][height][contentpos]{width} text \end{minipage}` environment.

The `pos` parameter can take one of the letters **c** enter, **t** op or **b** ottom to control the vertical alignment of the box, relative to the baseline of the surrounding text. The `height` parameter is the height of the parbox or minipage. The `contentpos` parameter is the position of the content and can be one of **c** enter, **t** op, **b** ottom or **s** pread. `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.

```
\noindent
\fbox{\parbox[b][4em][t]{0.33\textwidth}{Some \\ text} }
\fbox{\parbox[c][4em][s]{0.33\textwidth}{Some \vfill text} }
\fbox{\parbox[t][4em][c]{0.33\textwidth}{Some \\ text} }
```

This should print 3 boxes on the same line. Do not put another linebreak between the `\fbox`, otherwise you will put the following `\fbox` in another paragraph on another line.

25.6.1. Paragraphs in all boxes

You can make use of the *long* capabilities of minipage and parbox to embed paragraphs in non-long boxes. For instance:

```
\fbox{
  \parbox{\textwidth}{
    Some very long text...
  }
}
```

This prevents the overfull badness.

You can also use

```
\pbox{\textwidth}{my text}
```

from the pbox package which will create a box of minimal size around the text. Note that the \pbox command takes an optional argument that specifies the vertical position of the text:

```
\pbox[b]{\textwidth}{my text}
```

The valid values are b (bottom), t (top), and c (center). If you specify a length in the first (required) argument, the text will be wrapped:

```
\pbox[b]{5cm}{This is long text that will be wrapped once it reaches five
 centimeters.}
```

25.7. savebox

A `\savebox` is a reference to a box filled with contents. You can use it as a way to print or manipulate something repeatedly.

```
\newsavebox{\boxname}
\savebox{\boxname}{some content}
\usebox{\boxname}
```

The command `\newsavebox` creates a placeholder for storing a text; the command `\savebox` stores the specified text in this placeholder, and does not display anything in the document; and `\usebox` recalls the content of the placeholder into the document.

25.8. rotatebox

See Rotations[4].

25.9. colorbox and fcolorbox

See Colors[5]. `\fcolorbox` can also be tweaked with `\fboxsep` and `\fboxrule`.

25.10. resizebox and scalebox

The `graphicx` package feature additional boxes.

```
\resizebox{10ex}{2\baselineskip}{Dunhill style}
\scalebox{10}{Giant}
```

25.11. fancybox

the `fancybox` package provides additional boxes.

- `\doublebox`
- `\ovalbox`
- `\shadowbox`

4 Chapter 13 on page 151
5 Chapter 8 on page 89

26. Rules and Struts

26.1. Rules

The `\rule` command in normal use produces a simple black box:

```
\rule[depth]{width}{height}
```

The `depth`, `width` and `height` parameters are explained in the Boxes[1] chapter.

Here is an example:

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

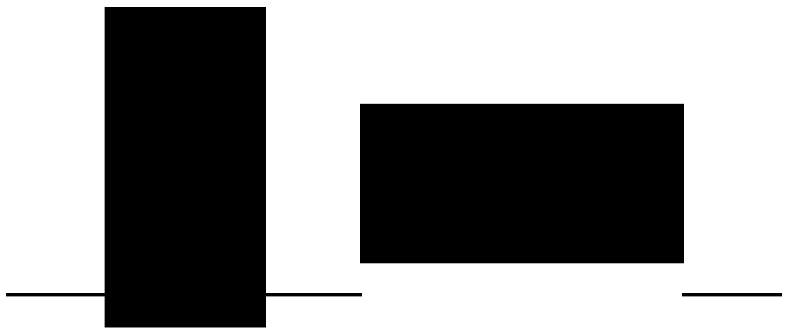


Figure 81

This is useful for drawing vertical and horizontal lines.

26.2. Struts

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 or in boxes to make sure a row has a certain minimum height.

1 Chapter 25.1 on page 295

In LaTeX a strut is defined as

```
\rule[-.3\baselineskip]{0pt}{\baselineskip}
```

26.3. Stretched rules

LaTeX provides the `\hrulefill` command, which work like a stretched horizontal space. See the Lengths[2] chapter.

2 Chapter 23 on page 283

Part IV.

Technical Texts

27. Mathematics

One of the greatest motivating forces for Donald Knuth when he began developing the original TeX system was to create something that allowed simple construction of mathematical formulas, while looking professional when printed. The fact that he succeeded was most probably why TeX (and later on, LaTeX) became so popular within the scientific community. Typesetting mathematics is one of LaTeX's greatest strengths. It is also a large topic due to the existence of so much mathematical notation.

If your document requires only a few simple mathematical formulas, plain LaTeX has most of the tools that you will need. If you are writing a scientific document that contains numerous complicated formulas, the `amsmath` package[1] introduces several new commands that are more powerful and flexible than the ones provided by LaTeX. The `mathtools` package fixes some `amsmath` quirks and adds some useful settings, symbols, and environments to amsmath.[2] To use either package, include:

```
\usepackage{amsmath}
```

or

```
\usepackage{mathtools}
```

in the preamble of the document. The `mathtools` package loads the `amsmath` package and hence there is no need to `\usepackage{amsmath}` in the preamble if `mathtools` is used.

27.1. Mathematics environments

LaTeX needs to know beforehand that the subsequent text does indeed contain mathematical elements. This is because LaTeX typesets maths notation differently from normal text. Therefore, special environments have been declared for this purpose. They can be distinguished into two categories depending on how they are presented:

- *text* — text formulas are displayed inline, that is, within the body of text where it is declared, for example, I can say that $a + a = 2a$ within this sentence.
- *displayed* — displayed formulas are separate from the main text.

As maths require special environments, there are naturally the appropriate environment names you can use in the standard way. Unlike most other environments, however, there are some handy shorthands to declaring your formulas. The following table summarizes them:

1 `http://www.ams.org/publications/authors/tex/amslatex`
2 `http://www.tex.ac.uk/ctan/macros/latex/contrib/mathtools/mathtools.pdf`

Type	**Inline (within text) formulas**	**Displayed equations**	**Displayed and automatically numbered equations**
Environment	`math`	`displaymath`	`equation`
LaTeX shorthand	`\(...\)`	`\[...\]`	
TeX shorthand	`$...$`	`$$...$$`	
Comment			`equation*` (starred version) suppresses numbering, but requires amsmath

Suggestion : Using the `$$...$$` should be avoided, as it may cause problems, particularly with the AMS-LaTeX macros. Furthermore, should a problem occur, the error messages may not be helpful.

The `equation*` and `displaymath` environments are functionally equivalent.

If you are typing text normally, you are said to be in *text mode* , but while you are typing within one of those mathematical environments, you are said to be in *math mode* , that has some differences compared to the *text mode* :

1. Most spaces and line breaks do not have any significance, as all spaces are either derived logically from the mathematical expressions, or have to be specified with special commands such as `\quad`
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 dedicated commands.[3]

27.1.1. Inserting "Displayed" maths inside blocks of text

In order for some operators, such as `\lim` or `\sum` to be displayed correctly inside some math environments (read `$......$`), it might be convenient to write the `\displaystyle` class inside the environment. Doing so might cause the line to be taller, but will cause exponents and indices to be displayed correctly for some math operators. For example, the `$\sum$` will print a smaller Σ and `$\displaystyle \sum$` will print a bigger one $\displaystyle\sum$, like in equations (This only works with AMSMATH package). It is also possible to force this behaviour for all math environments by declaring `\everymath{\displaystyle}` at the very beginning (i.e. before `\begin{document}`), which is useful in longer documents.

27.2. Symbols

Mathematics has many symbols! One of the most difficult aspects of learning LaTeX is remembering how to produce symbols. There is of course a set of symbols that can be accessed directly from the keyboard:

3 Chapter 27.11 on page 324

```
+ - = ! / ( ) [ ] < > | ' :
```

Beyond those listed above, distinct commands must be issued in order to display the desired symbols. There are a great deal of examples such as Greek letters, set and relations symbols, arrows, binary operators, etc.

For example:

```
\forall x \in X, \quad \exists y \leq \epsilon
```

$$\forall x \in X, \quad \exists y \leq \epsilon$$

Fortunately, there's a tool that can greatly simplify the search for the command for a specific symbol. Look for "Detexify" in the external links[4] section below. Another option would be to look in the "The Comprehensive LaTeX Symbol List" in the external links[5] section below.

27.3. Greek letters

Greek letters are commonly used in mathematics, and they are very easy to type in *math mode* . You just have to type the name of the letter after a backslash: if the first letter is lowercase, you will get a lowercase Greek letter, if the first letter is uppercase (and only the first letter), then you will get an uppercase letter. Note that some uppercase Greek letters look like Latin ones, so they are not provided by LaTeX (e.g. uppercase *Alpha* and *Beta* are just "A" and "B" respectively). Lowercase epsilon, theta, kappa, phi, pi, rho, and sigma are provided in two different versions. The alternate, or *var* iant, version is created by adding "var" before the name of the letter:

```
\alpha, \Alpha, \beta, \Beta, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \Phi
```

$$\alpha, \mathrm{A}, \beta, \mathrm{B}, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \Phi$$

Scroll down to #List of Mathematical Symbols[6] for a complete list of Greek symbols.

4 Chapter 27.22 on page 342
5 Chapter 27.22 on page 342
6 Chapter 27.18 on page 334

27.4. Operators

An operator is a function that is written as a word: e.g. trigonometric functions (sin, cos, tan), logarithms and exponentials (log, exp). LaTeX has many of these defined as commands:

```
\cos (2\theta) = \cos^2 \theta - \sin^2 \theta
```

$$\cos(2\theta) = \cos^2\theta - \sin^2\theta$$

For certain operators such as limits[7], the subscript is placed underneath the operator:

```
\lim_{x \to \infty} \exp(-x) = 0
```

$$\lim_{x\to\infty}\exp(-x) = 0$$

For the modular operator[8] there are two commands: `\bmod` and `\pmod`:

```
a \bmod b
```

$$a \bmod b$$

```
x \equiv a \pmod b
```

$$x \equiv a \pmod b$$

To use operators that are not pre-defined, such as argmax[9], see custom operators[10]

7 http://en.wikipedia.org/wiki/Limit%20%28mathematics%29
8 http://en.wikipedia.org/wiki/Modular%20arithmetic
9 http://en.wikipedia.org/wiki/argmax
10 Chapter 28.6 on page 358

27.5. Powers and indices

Powers and indices are equivalent to superscripts and subscripts in normal text mode. The caret (`^`) character is used to raise something, and the underscore (`_`) is for lowering. If more than one expression is raised or lowered, they should be grouped using curly braces (`{` and `}`).

```
k_{n+1} = n^2 + k_n^2 - k_{n-1}
```

$k_{n+1} = n^2 + k_n^2 - k_{n-1}$

For powers with more than one digit, surround the power with {}.

```
n^{22}
```

n^{22}

An underscore (`_`) can be used with a vertical bar (|) to denote evaluation using subscript notation in mathematics:

```
f(n) = n^5 + 4n^2 + 2 _{n=17}
```

$f(n) = n^5 + 4n^2 + 2|_{n=17}$

27.6. Fractions and Binomials

A fraction is created using the `\frac{numerator}{denominator}` command. (for those who need their memories refreshed, that's the *top* and *bottom* respectively!). Likewise, the binomial coefficient[11] (aka the Choose function) may be written using the `\binom` command[12]:

```
\frac{n!}{k!(n-k)!} = \binom{n}{k}
```

$\frac{n!}{k!(n-k)!} = \binom{n}{k}$

11 `http://en.wikipedia.org/wiki/Binomial%20coefficient`
12 requires the `amsmath` package

It is also possible to use the `\choose` command without the `amsmath` package:

```
\frac{n!}{k!(n-k)!} = {n \choose k}
```

$\frac{n!}{k!(n-k)!} = {n \choose k}$

You can embed fractions within fractions:

```
\frac{\frac{1}{x}+\frac{1}{y}<!---->}{y-z}
```

$\frac{\frac{1}{x}+\frac{1}{y}}{y-z}$

Note that when appearing inside another fraction, or in inline text $\frac{a}{b}$, a fraction is noticeably smaller than in displayed mathematics. The `\tfrac` and `\dfrac` commands[13] force the use of the respective styles, `\textstyle` and `\displaystyle`. Similarly, the `\tbinom` and `\dbinom` commands typeset the binomial coefficient.

Another way to write fractions is to use the `\over` command without the `amsmath` package:

```
{n! \over k!(n-k)!} = {n \choose k}
```

${n! \over k!(n-k)!} = {n \choose k}$

For relatively simple fractions, especially within the text, it may be more aesthetically pleasing to use powers and indices[14]:

```
^3/_7
```

${}^3/_7$

13 requires the `amsmath` package
14 Chapter 27.5 on page 309

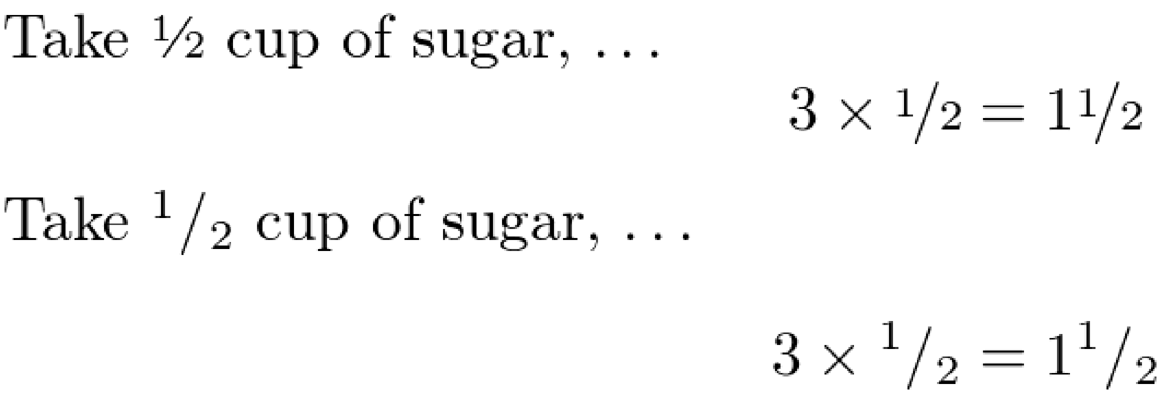
If this looks a little ”loose” (overspaced), a tightened version can be defined by inserting some negative space

```
%running fraction with slash - requires math mode.
\newcommand*\rfrac[2]

\rfrac{3}{7}
```

${}^3/_7$

If you use them throughout the document, usage of `xfrac` package is recommended. This package provides `\sfrac` command to create slanted fractions. Usage:

```
Take $\sfrac{1}{2}$ cup of sugar, \dots

  3\times\sfrac{1}{2}=1\sfrac{1}{2}


Take ${}^1/_2$ cup of sugar, \dots

  3\times{}^1/_2=1{}^1/_2
```


Take ½ cup of sugar, …

$$3 \times {}^1\!/\!{}_2 = 1{}^1\!/\!{}_2$$

Take ${}^1/_2$ cup of sugar, …

$$3 \times {}^1/_2 = 1{}^1/_2$$

Figure 82

If fractions are used as an exponent curly braces have to be used around the `\sfrac` command:

$x^\frac{1}{2}$ % no error $x^\sfrac{1}{2}$ % error $x^{\sfrac{1}{2}}$ % no error

```
  $x^\frac{1}{2}$ % no error
```

$x^{\frac{1}{2}}$

In some cases, using the package alone will result in errors about certain font shapes not being available. In that case, the `lmodern` and `fix-cm` packages need to be added as well.

Alternatively, the `nicefrac` package provides the `\nicefrac` command, whose usage is similar to `\sfrac`.

27.6.1. Continued fractions

Continued fractions should be written using `\cfrac` command[15]:

```
\begin{equation}
  x = a_0 + \cfrac{1}{a_1
          + \cfrac{1}{a_2
          + \cfrac{1}{a_3 + \cfrac{1}{a_4} } } }
\end{equation}
```

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}}$$

27.6.2. Multiplication of two numbers

To make multiplication visually similar to a fraction, a nested array can be used, for example multiplication of numbers written one below the other.

```
\begin{equation}
\frac{
    \begin{array}[b]{r}
      \left( x_1 x_2 \right)\\
      \times \left( x'_1 x'_2 \right)
    \end{array}
  }{
    \left( y_1y_2y_3y_4 \right)
  }
\end{equation}
```

$$\frac{\begin{array}[b]{r}\left(x_1 x_2 \right)\\ \times \left(x'_1 x'_2 \right)\end{array}}{\left(y_1y_2y_3y_4 \right)}$$

15 requires the `amsmath` package

27.7. Roots

The `\sqrt` command creates a square root surrounding an expression. It accepts an optional argument specified in square brackets (`[` and `]`) to change magnitude:

```
\sqrt{\frac{a}{b}}
```

$\sqrt{\frac{a}{b}}$

```
\sqrt[n]{1+x+x^2+x^3+\ldots}
```

$\sqrt[n]{1+x+x^2+x^3+\ldots}$

Some people prefer writing the square root "closing" it over its content. This method arguably makes it more clear what is in the scope of the root sign. This habit is not normally used while writing with the computer, but if you still want to change the output of the square root, LaTeX gives you this possibility. Just add the following code in the preamble of your document:

```
% New definition of square root:
% it renames \sqrt as \oldsqrt
\let\oldsqrt\sqrt
% it defines the new \sqrt in terms of the old one
\def\sqrt{\mathpalette\DHLhksqrt}
\def\DHLhksqrt#1#2{%
\setbox0=\hbox{$#1\oldsqrt{#2\,}$}\dimen0=\ht0
\advance\dimen0-0.2\ht0
\setbox2=\hbox{\vrule height\ht0 depth -\dimen0}%
{\box0\lower0.4pt\box2}}
```

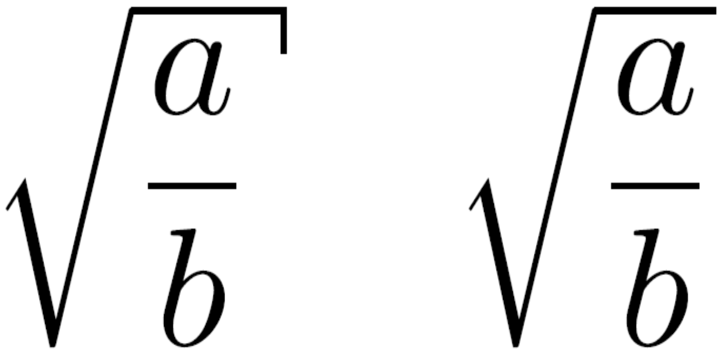


Figure 83 The new style is on left, the old one on right

This TeX code first renames the `\sqrt` command as `\oldsqrt`, then redefines `\sqrt` in terms of the old one, adding something more. The new square root can be seen in the picture on the left, compared to the old one on the right. Unfortunately this code won't work if you want to use multiple roots: if you try to write $\sqrt[b]{a}$ as `\sqrt[b]{a}` after you used the code above, you'll just get a wrong output. In other words, you can redefine the square root this way only if you are not going to use multiple roots in the whole document.

An alternative piece of TeX code that does allow multiple roots is

```
\usepackage{letltxmacro}
\makeatletter
\let\oldr@@t\r@@t
\def\r@@t#1#2{%
\setbox0=\hbox{$\oldr@@t#1{#2\,}$}\dimen0=\ht0
\advance\dimen0-0.2\ht0
\setbox2=\hbox{\vrule height\ht0 depth -\dimen0}%
{\box0\lower0.4pt\box2}}
\LetLtxMacro{\oldsqrt}{\sqrt}
\renewcommand*{\sqrt}[2][\ ]{\oldsqrt[#1]{#2} }
\makeatother


$\sqrt[a]{b} \quad \oldsqrt[a]{b}$
```

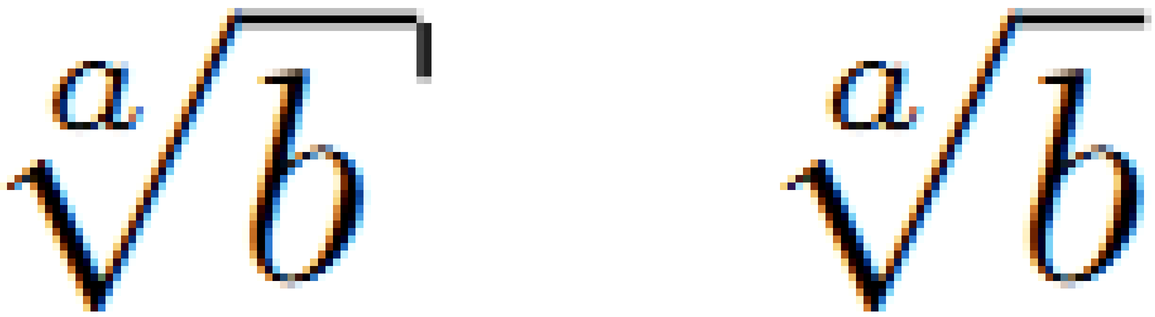


Figure 84

However this requires the `\usepackage{letltxmacro}` package

27.8. Sums and integrals

The `\sum` and `\int` commands insert the sum and integral symbols respectively, with limits specified using the caret (`^`) and underscore (`_`). The typical notation for sums is:

```
\sum_{i=1}^{10} t_i
```

$\sum_{i=1}^{10} t_i$

or

```
\displaystyle\sum_{i=1}^{10} t_i
```

$$\sum_{i=1}^{10} t_i$$

The limits for the integrals follow the same notation. It's also important to represent the integration variables with an upright d, which in math mode is obtained through the \mathrm{} command, and with a small space separating it from the integrand, which is attained with the \, command.

```
\int_0^\infty \mathrm{e}^{-x}\,\mathrm{d}x
```

$\int_0^\infty \mathrm{e}^{-x}\,\mathrm{d}x$

There are many other "big" commands which operate in a similar manner:

`\sum`	$\sum$	`\prod`	$\prod$	`\coprod`	
`\bigoplus`	$\bigoplus$	`\bigotimes`	$\bigotimes$	`\bigodot`	$\bigodot$
`\bigcup`	$\bigcup$	`\bigcap`	$\bigcap$	`\biguplus`	$\biguplus$
`\bigsqcup`	$\bigsqcup$	`\bigvee`	$\bigvee$	`\bigwedge`	$\bigwedge$
`\int`	$\int$	`\oint`	$\oint$	`\iint` [16]	$\iint$
`\iiint` [17]	$\iiint$	`\iiiint` [18]	$\iiiint$	`\idotsint` [19]	$\int\cdots\int$

16 requires the `amsmath` package
17 requires the `amsmath` package
18 requires the `amsmath` package
19 requires the `amsmath` package

For more integral symbols, including those not included by default in the Computer Modern font, try the `esint` package.

The `\substack` command[20] allows the use of `\\` to write the limits over multiple lines:

```
\sum_{\substack{
  0<i<m \\
  0<j<n
 }<!---->}
P(i,j)
```

$\sum_{\substack{0<i<m \\ 0<j<n}} P(i,j)$

If you want the limits of an integral to be specified above and below the symbol (like the sum), use the `\limits` command:

```
\int\limits_a^b
```

$\int\limits_a^b$

However if you want this to apply to ALL integrals, it is preferable to specify the `intlimits` option when loading the `amsmath` package:

```
\usepackage[intlimits]{amsmath}
```

Subscripts and superscripts in other contexts as well as other parameters to `amsmath` package related to them are described in Advanced Mathematics[21] chapter.

For bigger integrals, you may use personal declarations, or the `bigints` package [22].

27.9. Brackets, braces and delimiters

How to use braces in multi line equations is described in the Advanced Mathematics[23] chapter.

The use of delimiters such as brackets soon becomes important when dealing with anything but the most trivial equations. Without them, formulas can become ambiguous. Also,

20 requires the `amsmath` package
21 Chapter 28.7 on page 359
22 `http://hdl.handle.net/2268/6219`
23 Chapter 28.2.3 on page 350

special types of mathematical structures, such as matrices, typically rely on delimiters to enclose them.

There are a variety of delimiters available for use in LaTeX:

```
( a ), [ b ], \{ c \},  d , \ e \,
\langle f \rangle, \lfloor g \rfloor,
\lceil h \rceil, \ulcorner i \urcorner
```

$$(a),[b],\{c\},|d|,\|e\|,\langle f\rangle,\lfloor g\rfloor,\lceil h\rceil,\ulcorner i\urcorner$$

27.9.1. Automatic sizing

Very often mathematical features will differ in size, in which case the delimiters surrounding the expression should vary accordingly. This can be done automatically using the `\left`, `\right`, and `\middle` commands. Any of the previous delimiters may be used in combination with these:

```
\left(\frac{x^2}{y^3}\right)
```

$$\left(\frac{x^2}{y^3}\right)$$

```
P\left(A=2\middle\frac{A^2}{B}>4\right)
```

$$P\left(A=2\middle|\frac{A^2}{B}>4\right)$$

Figure 85

Curly braces are defined differently by using `\left\{` and `\right\}`,

```
\left\{\frac{x^2}{y^3}\right\}
```

$\left\{\frac{x^2}{y^3}\right\}$

If a delimiter on only one side of an expression is required, then an invisible delimiter on the other side may be denoted using a period (`.`).

```
\left.\frac{x^3}{3}\right_0^1
```

$\left.\frac{x^3}{3}\right|_0^1$

27.9.2. Manual sizing

In certain cases, the sizing produced by the `\left` and `\right` commands may not be desirable, or you may simply want finer control over the delimiter sizes. In this case, the `\big`, `\Big`, `\bigg` and `\Bigg` modifier commands may be used:

```
( \big( \Big( \bigg( \Bigg(
```

$(\big(\Big(\bigg(\Bigg($

These commands are primarily useful when dealing with nested delimiters. For example, when typesetting

```
\frac{\mathrm d}{\mathrm d x} \left( k g(x) \right)
```

$\frac{\mathrm d}{\mathrm d x} \left(k g(x) \right)$

we notice that the `\left` and `\right` commands produce the same size delimiters as those nested within it. This can be difficult to read. To fix this, we write

```
\frac{\mathrm d}{\mathrm d x} \big( k g(x) \big)
```

$\frac{\mathrm{d}}{\mathrm{d}x}(kg(x))$

Manual sizing can also be useful when an equation is too large, trails off the end of the page, and must be separated into two lines using an align command. `\left` and `\right` will give errors if the left and right appear on different lines.

27.9.3. Typesetting intervals

To denote open and half-open intervals, the notations]a,b[, (a,b),]a,b], (a,b], [a,b[and [a,b) are used. If the square bracket notation is used, then the interval must be put between curly braces (`{` and `}`) in order to have correct spacing. Similarly, if a (half-)open interval starts with a negative number, then the number including its minus-symbol must also be put between curly brackets, so that LaTeX understands that the minus-symbol is the unary operation. Compare:

```
x \in [-1,1]
```

$x \in [-1,1]$

```
x \in {[-1,1]}
```

$x \in {[-1,1]}$

```
x \in {[{-1},1]}
```

$x \in {[{-1},1]}$

27.10. Matrices and arrays

A basic matrix may be created using the `matrix` environment[24]: in common with other table-like structures, entries are specified by row, with columns separated using an ampersand (`&`) and a new rows separated with a double backslash (\\)

24 requires the `amsmath` package

```
\begin{matrix}
 a & b & c \\
 d & e & f \\
 g & h & i
\end{matrix}
```

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$

To specify alignment of columns in the table, use starred version[25]:

```
\begin{matrix}
 -1 & 3 \\
 2 & -4
\end{matrix}
=
\begin{matrix*}[r]
 -1 & 3 \\
 2 & -4
\end{matrix*}
```

$$\begin{matrix} -1 & 3 \\ 2 & -4 \end{matrix} = \begin{matrix*}[r] -1 & 3 \\ 2 & -4 \end{matrix*}$$

The alignment by default is `c` but it can be any column type valid in `array` environment.

However matrices are usually enclosed in delimiters of some kind, and while it is possible to use the `\left` and `\right` commands[26], there are various other predefined environments which automatically include delimiters:

Environment name	**Surrounding delimiter**	**Notes**
`pmatrix`[27]	( )	centers columns by default
`pmatrix*`[28]	( )	allows to specify alignment of columns in optional parameter
`bmatrix`[29]	[ ]	centers columns by default

25 requires the `mathtools` package
26 Chapter 27.9.1 on page 318
27 requires the `amsmath` package
28 requires the `mathtools` package
29 requires the `amsmath` package

Environment name	**Surrounding delimiter**	**Notes**
`bmatrix*`[30]	[]	allows to specify alignment of columns in optional parameter
`Bmatrix`[31]	{}	centers columns by default
`Bmatrix*`[32]	{}	allows to specify alignment of columns in optional parameter
`vmatrix`[33]	||	centers columns by default
`vmatrix*`[34]	||	allows to specify alignment of columns in optional parameter
`Vmatrix`[35]	‖ ‖	centers columns by default
`Vmatrix*`[36]	‖ ‖	allows to specify alignment of colums in optional parameter

When writing down arbitrary sized matrices, it is common to use horizontal, vertical and diagonal triplets of dots (known as ellipses[37]) to fill in certain columns and rows. These can be specified using the `\cdots`, `\vdots` and `\ddots` respectively:

```
A_{m,n} =
\begin{pmatrix}
 a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
 a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
 \vdots  & \vdots  & \ddots & \vdots  \\
 a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}
```

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

In some cases you may want to have finer control of the alignment within each column, or want to insert lines between columns or rows. This can be achieved using the `array`

30 requires the `mathtools` package
31 requires the `amsmath` package
32 requires the `mathtools` package
33 requires the `amsmath` package
34 requires the `mathtools` package
35 requires the `amsmath` package
36 requires the `mathtools` package
37 `http://en.wikipedia.org/wiki/ellipsis`

environment, which is essentially a math-mode version of the `tabular` environment[38], which requires that the columns be pre-specified:

```
\begin{array}{cc}
 1 & 2 \\
 \hline
 3 & 4
\end{array}
```

$$\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}$$

You may see that the AMS matrix class of environments doesn't leave enough space when used together with fractions resulting in output similar to this:

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

To counteract this problem, add additional leading space with the optional parameter to the \\ command:

```
M = \begin{bmatrix}
      \frac{5}{6} & \frac{1}{6} & 0           \\[0.3em]
      \frac{5}{6} & 0           & \frac{1}{6} \\[0.3em]
      0           & \frac{5}{6} & \frac{1}{6}
    \end{bmatrix}
```

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\[0.3em] \frac{5}{6} & 0 & \frac{1}{6} \\[0.3em] 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

If you need "border" or "indexes" on your matrix, plain TeX provides the macro `\bordermatrix`

```
M = \bordermatrix{~ & x & y \cr
                  A & 1 & 0 \cr
                  B & 0 & 1 \cr}
```

38 Chapter 14.6 on page 173

$$M = \begin{array}{cc} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} A \\ B \end{array} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

Figure 86

27.10.1. Matrices in running text

To insert a small matrix, and not increase leading in the line containing it, use `smallmatrix` environment:

```
A matrix in text must be set smaller:
$\bigl(\begin{smallmatrix}
a&b \\ c&d
\end{smallmatrix} \bigr)$
to not increase leading in a portion of text.
```

A matrix in text must be set smaller: $\bigl(\begin{smallmatrix} a&b \\ c&d \end{smallmatrix}\bigr)$ to not increase leading in a portion of text.

Figure 87

27.11. Adding text to equations

The math environment differs from the text environment in the representation of text. Here is an example of trying to represent text within the math environment:

```
50 apples \times 100 apples = lots of apples^2
```

$$50 apples \times 100 apples = lots of apples^2$$

There are two noticeable problems: there are no spaces between words or numbers, and the letters are italicized and more spaced out than normal. Both issues are simply artifacts of the maths mode, in that it treats it as a mathematical expression: spaces are ignored

(LaTeX spaces mathematics according to its own rules), and each character is a separate element (so are not positioned as closely as normal text).

There are a number of ways that text can be added properly. The typical way is to wrap the text with the `\text{...}` command [39] (a similar command is `\mbox{...}`, though this causes problems with subscripts, and has a less descriptive name). Let's see what happens when the above equation code is adapted:

```
50 \text{apples} \times 100 \text{apples}
= \text{lots of apples}^2
```

$$50\text{apples} \times 100\text{apples} = \text{lots of apples}^2$$

The text looks better. However, there are no gaps between the numbers and the words. Unfortunately, you are required to explicitly add these. There are many ways to add spaces between maths elements, but for the sake of simplicity we may simply insert space characters into the `\text` commands.

```
50 \text{ apples} \times 100 \text{ apples}
= \text{lots of apples}^2
```

$$50\text{ apples} \times 100\text{ apples} = \text{lots of apples}^2$$

27.11.1. Formatted text

Using the `\text` is fine and gets the basic result. Yet, there is an alternative that offers a little more flexibility. You may recall the introduction of font formatting commands[40], such as `\textrm`, `\textit`, `\textbf`, etc. These commands format the argument accordingly, e.g., `\textbf{bold text}` gives **bold text** . These commands are equally valid within a maths environment to include text. The added benefit here is that you can have better control over the font formatting, rather than the standard text achieved with `\text`.

```
50 \textrm{ apples} \times 100
\textbf{ apples} = \textit{lots of apples}^2
```

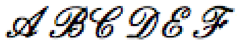
39 requires the `amsmath` package
40 Chapter 6.10 on page 73

$50 \text{ apples} \times 100 \textbf{ apples} = \textit{lots of apples}^2$

27.12. Formatting mathematics symbols

See also: w:Mathematical Alphanumeric Symbols[41]*, w:Help:Displaying a formula#Alphabets and typefaces*[42] *and w:Wikipedia:LaTeX symbols#Fonts*[43]

We can now format text; what about formatting mathematical expressions? There are a set of formatting commands very similar to the font formatting ones just used, except that they are specifically aimed at text in math mode (requires `amsfonts`)

LaTeX command	Sample	Description	Common use
`\mathnormal{…}` (or simply omit any command)	$ABCDEF\ abcdef\ 123456$	The default math font	Most mathematical notation
`\mathrm{…}`	$\mathrm{ABCDEF\ abcdef\ 123456}$	This is the default or normal font, unitalicised	Units of measurement, one word functions
`\mathit{…}`	$\mathit{ABCDEF\ abcdef\ 123456}$	Italicised font	Multi-letter function or variable names. Compared to `\mathnormal` , words are spaced more naturally and numbers are italicized as well.
`\mathbf{…}`	$\mathbf{ABCDEF\ abcdef\ 123456}$	Bold font	Vectors
`\mathsf{…}`	$\mathsf{ABCDEF\ abcdef\ 123456}$	Sans-serif[44]	
`\mathtt{…}`	$\mathtt{ABCDEF\ abcdef\ 123456}$	Monospace (fixed-width) font[45]	
`\mathfrak{…}`	$\mathfrak{ABCDEF\ abcdef\ 123456}$	Fraktur[46]	Almost canonical font for Lie algebras, with superscript used to denote New Testament papyri[47], ideals[48] in ring theory
`\mathcal{…}`	$\mathcal{ABCDEF}$	Calligraphy (uppercase only)	Often used for sheaves/schemes and categories, used to denote cryptological[49] concepts like an *alphabet of definition* ($\mathcal{A}$), *message space* ($\mathcal{M}$), *ciphertext space* ($\mathcal{C}$) and *key space*[50] ($\mathcal{K}$); Kleene's $\mathcal{O}$[51]; naming convention in description logic[52]; Laplace transform[53] ($\mathcal{L}$) and Fourier transform[54] ($\mathcal{F}$)
`\mathbb{…}` (requires the `amsfonts` or `amssymb` package)	$\mathbb{ABCDEF}$	Blackboard bold[55] (uppercase only)	Used to denote special sets (e.g. real numbers)
`\mathscr{…}` (requires the `mathrsfs` package)	 **Figure 88**	Script[56] (uppercase only)	An alternative font for categories and sheaves.

These formatting commands can be wrapped around the entire equation, and not just on the textual elements: they only format letters, numbers, and uppercase Greek, and other math commands are unaffected.

41 http://en.wikipedia.org/wiki/Mathematical%20Alphanumeric%20Symbols
42 http://en.wikipedia.org/wiki/Help%3ADisplaying%20a%20formula%23Alphabets%20and%20typefaces
43 http://en.wikipedia.org/wiki/Wikipedia%3ALaTeX%20symbols%23Fonts
44 http://en.wikipedia.org/wiki/sans-serif
45 http://en.wikipedia.org/wiki/Monospace%20font
46 http://en.wikipedia.org/wiki/Fraktur%20%28script%29
47 http://en.wikipedia.org/wiki/List%20of%20New%20Testament%20papyri
48 http://en.wikipedia.org/wiki/Ideal%20%28ring%20theory%29
49 http://en.wikipedia.org/wiki/Cryptography
50 http://en.wikipedia.org/wiki/key%20space
51 http://en.wikipedia.org/wiki/Kleene%27s%20O
52 http://en.wikipedia.org/wiki/Description%20logic%23Naming%20Convention
53 http://en.wikipedia.org/wiki/Laplace%20transform
54 http://en.wikipedia.org/wiki/Fourier%20transform
55 http://en.wikipedia.org/wiki/Blackboard%20bold
56 http://en.wikipedia.org/wiki/Script%20%28typefaces%29

To bold lowercase Greek or other symbols use the `\boldsymbol` command[57]; this will only work if there exists a bold version of the symbol in the current font. As a last resort there is the `\pmb` command[58] (poor mans bold): this prints multiple versions of the character slightly offset against each other.

```
\boldsymbol{\beta} = (\beta_1,\beta_2,\dotsc,\beta_n)
```

$$\boldsymbol{\beta} = (\beta_1,\beta_2,\dotsc,\beta_n)$$

To change the size of the fonts in math mode, see Changing font size[59].

27.12.1. Accents

So what to do when you run out of symbols and fonts? Well the next step is to use accents:

`a' or a^{\prime}`	a'	`a''`	a''
`\hat{a}`	$\hat{a}$	`\bar{a}`	$\bar{a}$
`\grave{a}`	$\grave{a}$	`\acute{a}`	$\acute{a}$
`\dot{a}`	$\dot{a}$	`\ddot{a}`	$\ddot{a}$
`\not{a}`	$\not{a}$	`\mathring{a}`	
`\overrightarrow{AB}`	$\overrightarrow{AB}$	`\overleftarrow{AB}`	$\overleftarrow{AB}$
`a'''`	a'''	`a''''`	a''''
`\overline{aaa}`	$\overline{aaa}$	`\check{a}`	$\check{a}$
`\breve{a}`	$\breve{a}$	`\vec{a}`	$\vec{a}$
`\dddot{a}` [60]		`\ddddot{a}` [61]	
`\widehat{AAA}`	$\widehat{AAA}$	`\widetilde{AAA}`	$\widetilde{AAA}$
`\widehat{AAA}`	$\widehat{AAA}$	`\stackrel\frown{AAA}`	$\stackrel\frown{AAA}$
`\tilde{a}`		`\underline{a}`	$\underline{a}$

27.13. Color

The package `xcolor`, described in Colors[62], allows us to add color to our equations. For example,

57 requires the `amsmath` package
58 requires the `amsmath` package
59 Chapter 28.9 on page 363
60 requires the `amsmath` package
61 requires the `amsmath` package
62 Chapter 8.1 on page 89

```
k = {\color{red}x} \mathbin{\color{blue}-} 2
```

$k = {\color{red}x} \mathbin{\color{blue}-} 2$

The only problem is that this disrupts the default LaTeX formatting around the `-` operator. To fix this, we enclose it in a `\mathbin` environment, since `-` is a binary operator. This process is described here[63].

27.14. Plus and minus signs

Latex deals with the $+$ and $-$ signs in two possible ways. The most common is as a binary operator. When two maths elements appear on either side of the sign, it is assumed to be a binary operator, and as such, allocates some space either side of the sign. The alternative way is a sign designation. This is when you state whether a mathematical quantity is either positive or negative. This is common for the latter, as in maths, such elements are assumed to be positive unless a $-$ is prefixed to it. In this instance, you want the sign to appear close to the appropriate element to show their association. If you put a $+$ or a $-$ with nothing before it but you want it to be handled like a binary operator you can add an *invisible* character before the operator using `{}`. This can be useful if you are writing multiple-line formulas, and a new line could start with a $=$ or a $+$, for example, then you can fix some strange alignments adding the invisible character where necessary.

A plus-minus sign is written as:

```
\pm
```

$\pm$

Similarly, there exists also a minus-plus sign:

```
\mp
```

$\mp$

63 `http://tex.stackexchange.com/questions/21598/how-to-color-math-symbols`

27.15. Controlling horizontal spacing

LaTeX is obviously pretty good at typesetting maths—it was one of the chief aims of the core TeX system that LaTeX extends. However, it can't always be relied upon to accurately interpret formulas in the way you did. It has to make certain assumptions when there are ambiguous expressions. The result tends to be slightly incorrect horizontal spacing. In these events, the output is still satisfactory, yet any perfectionists will no doubt wish to *fine-tune* their formulas to ensure spacing is correct. These are generally very subtle adjustments.

There are other occasions where LaTeX has done its job correctly, but you just want to add some space, maybe to add a comment of some kind. For example, in the following equation, it is preferable to ensure there is a decent amount of space between the maths and the text.

```
\[ f(n) =
  \begin{cases}
    n/2       & \quad \text{if } n \text{ is even}\\
    -(n+1)/2  & \quad \text{if } n \text{ is odd}\\
  \end{cases}
\]
```

$$f(n) = \begin{cases} n/2 & \quad \text{if } n \text{ is even} \\ -(n+1)/2 & \quad \text{if } n \text{ is odd} \end{cases}$$

This code produces errors with Miktex 2.9 and does not yield the results seen on the right. Use \mathrm instead of just \text.

(Note that this particular example can be expressed in more elegant code by the `cases` construct provided by the `amsmath` package described in Advanced Mathematics[64] chapter.)

LaTeX has defined two commands that can be used anywhere in documents (not just maths) to insert some horizontal space. They are `\quad` and `\qquad`

A `\quad` is a space equal to the current font size. So, if you are using an 11pt font, then the space provided by `\quad` will also be 11pt (horizontally, of course.) The `\qquad` gives twice that amount. As you can see from the code from the above example, `\quad`s were used to add some separation between the maths and the text.

OK, so back to the fine tuning as mentioned at the beginning of the document. A good example would be displaying the simple equation for the indefinite integral of y with respect to x :

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

If you were to try this, you may write:

64 Chapter 28.2.2 on page 349

```
\int y \mathrm{d}x
```

$\int y \mathrm{d}x$

However, this doesn't give the correct result. LaTeX doesn't respect the white-space left in the code to signify that the y and the dx are independent entities. Instead, it lumps them altogether. A `\quad` would clearly be overkill in this situation—what is needed are some small spaces to be utilized in this type of instance, and that's what LaTeX provides:

Command	**Description**	**Size**
`\,`	small space	3/18 of a quad
`\:`	medium space	4/18 of a quad
`\;`	large space	5/18 of a quad
`\!`	negative space	-3/18 of a quad

NB you can use more than one command in a sequence to achieve a greater space if necessary.

So, to rectify the current problem:

```
\int y\, \mathrm{d}x
```

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

```
\int y\: \mathrm{d}x
```

$\int y \mathrm{d}x$

```
\int y\; \mathrm{d}x
```

$\int y \mathrm{d}x$

The negative space may seem like an odd thing to use, however, it wouldn't be there if it didn't have *some* use! Take the following example:

```
\left(
  \begin{array}{c}
    n \\
    r
  \end{array}
\right) = \frac{n!}{r!(n-r)!}
```

$$\left(\begin{array}{c} n \\ r \end{array}\right) = \frac{n!}{r!(n-r)!}$$

The matrix-like expression for representing binomial coefficients is too padded. There is too much space between the brackets and the actual contents within. This can easily be corrected by adding a few negative spaces after the left bracket and before the right bracket.

```
\left(\!
  \begin{array}{c}
    n \\
    r
  \end{array}
\!\right) = \frac{n!}{r!(n-r)!}
```

$$\left(\!\begin{array}{c} n \\ r \end{array}\!\right) = \frac{n!}{r!(n-r)!}$$

In any case, adding some spaces manually should be avoided whenever possible: it makes the source code more complex and it's against the basic principles of a What You See is What You Mean approach. The best thing to do is to define some commands using all the spaces you want and then, when you use your command, you don't have to add any other space. Later, if you change your mind about the length of the horizontal space, you can easily change it modifying only the command you defined before. Let us use an example: you want the *d* of a *dx* in an integral to be in roman font and a small space away from the rest. If you want to type an integral like `\int x \, \mathrm{d} x`, you can define a command like this:

```
\newcommand{\dd}{\mathop{}\,\mathrm{d}}
```

in the preamble of your document. We have chosen `\dd` just because it reminds the "d" it replaces and it is fast to type. Doing so, the code for your integral becomes `\int x \dd x`. Now, whenever you write an integral, you just have to use the `\dd` instead of the "d", and all your integrals will have the same style. If you change your mind, you just have to change the definition in the preamble, and all your integrals will be changed accordingly.

27.16. Manually Specifying Formula Style

To manually display a fragment of a formula using text style, surround the fragment with curly braces and prefix the fragment with `\textstyle`. The braces are required because the `\textstyle` macro changes the state of the renderer, rendering all subsequent mathematics in text style. The braces limit this change of state to just the fragment enclosed within. For example, to use text style for just the summation symbol in a sum, one would enter

```
\begin{equation}
   C^i_j = {\textstyle \sum_k} A^i_k B^k_j
\end{equation}
```

The same thing as a command would look like this:

```
\newcommand{\tsum}[1]}
```

Note the extra braces. Just one set around the expression won't be enough. That would cause all math after `\tsum k` to be displayed using text style.

To display part of a formula using display style, do the same thing, but use `\displaystyle` instead.

27.17. Advanced Mathematics: AMS Math package

The AMS (American Mathematical Society[65]) mathematics package is a powerful package that creates a higher layer of abstraction over mathematical LaTeX language; if you use it it will make your life easier. Some commands `amsmath` introduces will make other plain LaTeX commands obsolete: in order to keep consistency in the final output you'd better use `amsmath` commands whenever possible. If you do so, you will get an elegant output without worrying about alignment and other details, keeping your source code readable. If you want to use it, you have to add this in the preamble:

```
\usepackage{amsmath}
```

27.17.1. Introducing dots in formulas

`amsmath` defines also the `\dots` command, that is a generalization of the existing `\ldots`. You can use `\dots` in both text and math mode and LaTeX will replace it with three dots "..." but it will decide according to the context whether to put it on the bottom (like `\ldots`) or centered (like `\cdots`).

27.17.2. Dots

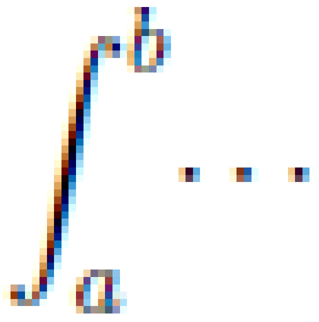
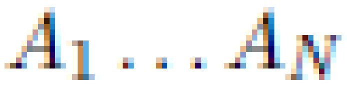
LaTeX gives you several commands to insert dots in your formulae. This can be particularly useful if you have to type big matrices omitting elements. First of all, here are the main dots-related commands LaTeX provides:

65 `http://en.wikipedia.org/wiki/American%20Mathematical%20Society`

Code	Output	Comment
`\dots`	$\dots$	generic dots, to be used in text (outside formulae as well). It automatically manages whitespaces before and after itself according to the context, it's a higher level command.
`\ldots`	$\ldots$	the output is similar to the previous one, but there is no automatic whitespace management; it works at a lower level.
`\cdots`	$\cdots$	These dots are centered relative to the height of a letter. There is also the binary multiplication operator, `\cdot` , mentioned below.
`\vdots`	$\vdots$	vertical dots
`\ddots`	$\ddots$	diagonal dots
`\iddots`		inverse diagonal dots (requires the `mathdots` package)
`\hdotsfor{n}`	$\dots\dots$	to be used in matrices, it creates a row of dots spanning n columns.

Instead of using `\ldots` and `\cdots`, you should use the semantically oriented commands. It makes it possible to adapt your document to different conventions on the fly, in case (for example) you have to submit it to a publisher who insists on following house tradition in this respect. The default treatment for the various kinds follows American Mathematical Society conventions.

Code	Output	Comment
`A_1,A_2,\dotsc,`	$A_1, A_2, \dots,$ **Figure 89**	for "dots with commas"
`A_1+\dotsb+A_N`	$A_1 + \cdots + A_N$ **Figure 90**	for "dots with binary operators/relations"
`A_1 \dotsm A_N`	$A_1 \cdots A_N$ **Figure 91**	for "multiplication dots"

Code	Output	Comment
`\int_a^b \dotsi`	 Figure 92	for ”dots with integrals”
`A_1\dotso A_N`	 Figure 93	for ”other dots” (none of the above)

27.17.3. Write an equation with the align environment

How to write an equation with the align environment with the `amsmath` package is described in Advanced Mathematics[66].

27.18. List of Mathematical Symbols

All the pre-defined mathematical symbols from the \TeX\ package are listed below. More symbols are available from extra packages.

66 Chapter 28.2.2 on page 349

Relation Symbols

Symbol	Script		Symbol	Script		Symbol	Script		Symbol	Script		Symbol	Script
$<$	`<`		$>$	`>`		$=$	`=`		$\parallel$	`\parallel`		$\nparallel$	`\nparallel`
$\leq$	`\leq`		$\geq$	`\geq`		$\doteq$	`\doteq`		$\asymp$	`\asymp`		$\bowtie$	`\bowtie`
$\ll$	`\ll`		$\gg$	`\gg`		$\equiv$	`\equiv`		$\vdash$	`\vdash`		$\dashv$	`\dashv`
$\subset$	`\subset`		$\supset$	`\supset`		$\approx$	`\approx`		$\in$	`\in`		$\ni$	`\ni`
$\subseteq$	`\subseteq`		$\supseteq$	`\supseteq`		$\cong$	`\cong`		$\smile$	`\smile`		$\frown$	`\frown`
$\nsubseteq$	`\nsubseteq`		$\nsupseteq$	`\nsupseteq`		$\simeq$	`\simeq`		$\models$	`\models`		$\notin$	`\notin`
$\sqsubset$	`\sqsubset`		$\sqsupset$	`\sqsupset`		$\sim$	`\sim`		$\perp$	`\perp`		$\mid$	`\mid`
$\sqsubseteq$	`\sqsubseteq`		$\sqsupseteq$	`\sqsupseteq`		$\propto$	`\propto`		$\prec$	`\prec`		$\succ$	`\succ`
$\preceq$	`\preceq`		$\succeq$	`\succeq`		$\neq$	`\neq`		$\sphericalangle$	`\spheri-calangle`		$\measuredangle$	`\mea-suredangle`

Binary Operations							
Symbol	**Script**	**Symbol**	**Script**	**Symbol**	**Script**	**Symbol**	**Script**
$\pm$	`\pm`	$\cap$	`\cap`	$\diamond$	`\diamond`	$\oplus$	`\oplus`
$\mp$	`\mp`	$\cup$	`\cup`	$\bigtriangleup$	`\bigtri- angleup`	$\ominus$	`\ominus`
$\times$	`\times`	$\uplus$	`\uplus`	$\bigtriangledown$	`\bigtri- angledown`	$\otimes$	`\otimes`
$\div$	`\div`	$\sqcap$	`\sqcap`	$\triangleleft$	`\trian- gleleft`	$\oslash$	`\oslash`
$\ast$	`\ast`	$\sqcup$	`\sqcup`	$\triangleright$	`\triang- leright`	$\odot$	`\odot`
$\star$	`\star`	$\vee$	`\vee`	$\bigcirc$	`\bigcirc`	$\circ$	`\circ`
$\dagger$	`\dagger`	$\wedge$	`\wedge`	$\bullet$	`\bullet`	$\setminus$	`\setminus`
$\ddagger$	`\ddagger`	$\cdot$	`\cdot`	$\wr$	`\wr`		`\amalg`

Set and/or Logic Notation				
Symbol	**Script**		**Symbol**	**Script**
$\exists$	`\exists`		$\rightarrow$	`\rightarrow` or `\to`
$\nexists$	`\nexists`		$\leftarrow$	`\leftarrow` or `\gets`
$\forall$	`\forall`		$\mapsto$	`\mapsto`
$\neg$	`\neg`		$\implies$	`\implies`
$\subset$	`\subset`		$\Rightarrow$	`\Rightarrow` (preferred for implication)
$\supset$	`\supset`		$\leftrightarrow$	`\leftrightarrow`
$\in$	`\in`		$\iff$	`\iff`
$\notin$	`\notin`		$\Leftrightarrow$	`\Leftrightarrow` (preferred for equivalence (iff))
$\ni$	`\ni`		$\top$	`\top`
$\land$	`\land`		$\bot$	`\bot`
$\lor$	`\lor`		$\emptyset$ and $\varnothing$	`\emptyset` and `\varnothing`

Delimiters

Symbol	Script		Symbol	Script		Symbol	Script		Symbol	Script
$\vert$	`|`		$\Vert$	`\\|`		$/$	`/`		$\backslash$	`\backslash`
$\{$	`\{`		$\}$	`\}`		$\langle$	`\langle`		$\rangle$	`\rangle`
$\uparrow$	`\uparrow`		$\Uparrow$	`\Uparrow`		$\lceil$	`\lceil`		$\rceil$	`\rceil`
$\downarrow$	`\downar- row`		$\Downarrow$	`\Downar- row`		$\lfloor$	`\lfloor`		$\rfloor$	`\rfloor`

Note: To use the Greek Letters in LaTeX that have the same appearance as their Roman equivalent, just use the Roman form: e.g., A instead of Alpha, B instead of Beta, etc.

Greek Letters				
Symbol	**Script**		**Symbol**	**Script**
A and α	`A` and `\alpha`		N and ν	`N` and `\nu`
B and β	`B` and `\beta`		Ξ and ξ	`\Xi` and `\xi`
Γ and γ	`\Gamma` and `\gamma`		and o	`O` and `o`
Δ and δ	`\Delta` and `\delta`		Π, π and ϖ	`\Pi` , `\pi` and `\varpi`
E, ϵ and ε	`E` , `\epsilon` and `\varepsilon`		P, ρ and ϱ	`P` , `\rho` and `\varrho`
Z and ζ	`Z` and `\zeta`		Σ, σ and ς	`\Sigma` , `\sigma` and `\var-sigma`
H and η	`H` and `\eta`		T and τ	`T` and `\tau`
Θ, θ and ϑ	`\Theta` , `\theta` and `\var-theta`		Υ and υ	`\Upsilon` and `\upsilon`
I and ι	`I` and `\iota`		Φ, ϕ, and φ	`\Phi` , `\phi` and `\varphi`
K, κ and $\varkappa$	`K` , `\kappa` and `\varkappa`		X and χ	`X` and `\chi`
Λ and λ	`\Lambda` and `\lambda`		Ψ and ψ	`\Psi` and `\psi`
M and μ	`M` and `\mu`		Ω and ω	`\Omega` and `\omega`

Other symbols

Symbol	**Script**	**Symbol**	**Script**	**Symbol**	**Script**	**Symbol**	**Script**	**Symbol**	**Script**
∂	`\partial`	$\imath$	`\imath`	$\Re$	`\Re`	∇	`\nabla`	$\aleph$	`\aleph`
ð	`\eth`		`\jmath`	$\Im$	`\Im`	$\Box$	`\Box`	ℶ	`\beth`
$\hbar$	`\hbar`	ℓ	`\ell`	$\wp$	`\wp`	∞	`\infty`	ℷ	`\gimel`

Trigonometric Functions

Symbol	**Script**		**Symbol**	**Script**		**Symbol**	**Script**		**Symbol**	**Script**
sin	`\sin`		arcsin	`\arcsin`		sinh	`\sinh`		sec	`\sec`
cos	`\cos`		arccos	`\arccos`		cosh	`\cosh`		csc	`\csc`
tan	`\tan`		arctan	`\arctan`		tanh	`\tanh`			
cot	`\cot`			`\arccot`		coth	`\coth`			

27.19. Summary

As you begin to see, typesetting math can be tricky at times. However, because LaTeX provides so much control, you can get professional quality mathematics typesetting with relatively little effort (once you've had a bit of practice, of course!). It would be possible to keep going and going with math topics because it seems potentially limitless. However, with this tutorial, you should be able to get along sufficiently.

27.20. Notes

27.21. Further reading

- meta:Help:Displaying a formula[67]: Wikimedia uses a subset of LaTeX commands.

27.22. External links

- LaTeX maths symbols[68]
- detexify[69]: applet for looking up LaTeX symbols by drawing them
- [ftp://ftp.ams.org/pub/tex/doc/amsmath/amsldoc.pdf `amsmath` documentation]
- LaTeX - The Student Room[70]
- The Comprehensive LaTeX Symbol List[71]
- MathLex - LaTeX math translator and equation builder[72]

pl:LaTeX/Matematyka[73]

67 `http://en.meta.org/wiki/Help%3ADisplaying%20a%20formula`
68 `http://www.artofproblemsolving.com/Wiki/index.php/LaTeX:Symbols`
69 `http://detexify.kirelabs.org`
70 `http://www.thestudentroom.co.uk/wiki/LaTeX`
71 `http://www.ctan.org/tex-archive/info/symbols/comprehensive`
72 `http://mathlex.org/latex`
73 `http://pl.wikibooks.org/wiki/LaTeX%2FMatematyka`

28. Advanced Mathematics

This page outlines some more advanced uses of mathematics markup using LaTeX. In particular it makes heavy use of the AMS-LaTeX packages supplied by the American Mathematical Society[1].

28.1. Equation numbering

The `equation` environment automatically numbers your equation:

```
\begin{equation}
 f(x)=(x+a)(x+b)
\end{equation}
```

$$f(x) = (x+a)(x+b) \qquad (1)$$

You can also use the `\label` and `\ref` (or `\eqref` from the `amsmath` package) commands to label and reference equations, respectively. For equation number 1, `\ref` results in 1 and `\eqref` results in (1):

```
\begin{equation} \label{eq:someequation}
5^2 - 5 = 20
\end{equation}

this references the equation \ref{eq:someequation}.
```

$$5^2 - 5 = 20 \qquad (1)$$

this references equation 1.

```
\begin{equation} \label{eq:erl}
a = bq + r
\end{equation}

where \eqref{eq:erl} is true if $a$ and $b$ are integers with $b \neq c$.
```

1 `http://en.wikipedia.org/wiki/American%20Mathematical%20Society`

$$a = bq + r \qquad (1)$$

where (1) is true if a and b
are integers with $b \neq c$.

Further information is provided in the labels and cross-referencing[2] chapter.

To have the enumeration follow from your section or subsection heading, you must use the `amsmath` package or use AMS class documents. Then enter

```
\numberwithin{equation}{section}
```

to the preamble to get enumeration at the section level or

```
\numberwithin{equation}{subsection}
```

to have the enumeration go to the subsection level.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
 \numberwithin{equation}{subsection}
 \begin{document}
 \section{First Section}

 \subsection{A subsection}
 \begin{equation}
  L' = {L}{\sqrt{1-\frac{v^2}{c^2}<!-- -->}<!-- -->}
 \end{equation}
\end{document}
```

$$L' = L\sqrt{1-\frac{v^2}{c^2}} \qquad (1.1.1)$$

If the style you follow requires putting dots after ordinals (as it is required at least in Polish typography) the `\numberwithin{equation}{subsection}` command in the preamble will result in the equation number in the above example to be rendered in this way: (1.1..1).

To remove the duplicate dot, add the following command immediately after `\numberwithin{equation}{section}`:

```
\renewcommand{\theequation}{\thesection\arabic{equation}}
```

For a numbering scheme using `\numberwithin{equation}{subsection}`, use:

```
\renewcommand{\theequation}{\thesubsection\arabic{equation}}
```

in the preamble of the document.

Note: Though it may look like the `\renewcommand` works by itself, it won't reset the equation number with each new section. It must be used together with manual equation number resetting after each new section beginning or with the much cleaner `\numberwithin`.

2 Chapter 21 on page 267

28.1.1. Subordinate equation numbering

To number subordinate equations in a numbered equation environment, place the part of document containing them in a `subequations` environment:

```
\begin{subequations}
Maxwell's equations:
\begin{align}
        B'&=-\nabla \times E,\\
        E'&=\nabla \times B - 4\pi j,
\end{align}
\end{subequations}
```

Maxwell's equations:

$$B' = -\nabla \times E, \qquad (1.1a)$$

$$E' = \nabla \times B - 4\pi j, \qquad (1.1b)$$

Referencing subordinate equations can be done using either of two methods: adding a label after the `\begin{subequations}` command, which will reference the main equation (1.1 above), or adding a label at the end of each line, before the `\\` command, which will reference the sub-equation (1.1a or 1.1b above). It is possible to add both labels in case both types of references are needed.

28.2. Vertically aligning displayed mathematics

An often encountered problem with displayed environments (`displaymath` and `equation`) is the lack of any ability to span multiple lines. While it is possible to define lines individually, these will not be aligned.

28.2.1. Above and below

The `\overset` and `\underset` commands[3] typeset symbols above and below expressions. Without AmsTex the same result of `\overset` can be obtained with `\stackrel`. This can be particularly useful for creating new binary relations:

```
\[
 A \overset{!}{=} B; A \stackrel{!}{=} B
\]
```

$$A \overset{!}{=} B; \quad A \stackrel{!}{=} B$$

3 requires the `amsmath` package

or to show usage of L'Hôpital's rule[4]:

```
\[
 \lim_{x\to 0}{\frac{e^x-1}{2x}<!-- -->}
 \overset{\left[\frac{0}{0}\right]}{\underset{\mathrm{H}<!-- -->}{=}<!-- -->}
 \lim_{x\to 0}{\frac{e^x}{2}<!-- -->}={\frac{1}{2}<!-- -->}
\]
```

$$\lim_{x\to 0}\tfrac{e^x-1}{2x} \overset{\left[\frac{0}{0}\right]}{\underset{\mathrm{H}}{=}} \lim_{x\to 0}\tfrac{e^x}{2}=\tfrac{1}{2}$$

It is convenient to define a new operator that will set the equal sign with H and the provided fraction:

```
\newcommand{\Heq}[1]{\overset{\left[#1\right]}{\underset{\mathrm{H}}{=}}}
```

which reduces the above example to:

```
\[
 \lim_{x\to 0}{\frac{e^x-1}{2x}}
 \Heq{\frac{0}{0}}
 \lim_{x\to 0}{\frac{e^x}{2}}={\frac{1}{2}}
\]
```

If the purpose is to make comments on particular parts of an equation, the `\overbrace` and `\underbrace` commands may be more useful. However, they have a different syntax (and can be aligned with the `\vphantom` command):

```
\[
 z = \overbrace{
   \underbrace{x}_\text{real} + i
   \underbrace{y}_\text{imaginary}
  }^\text{complex number}
\]
```

$$z = \overbrace{\underbrace{x}_\text{real} + i\underbrace{y}_\text{imaginary}}^\text{complex number}$$

Sometimes the comments are longer than the formula being commented on, which can cause spacing problems. These can be removed using the `\mathclap` command[5]:

4 http://en.wikipedia.org/wiki/L%27H%C3%B4pital%27s_rule

5 requires the `mathtools` package

```
\[
 y = a + f(\underbrace{b x}_{
                   \ge 0 \text{ by assumption}<!-- -->})
   = a + f(\underbrace{b x}_{
         \mathclap{\ge 0 \text{ by assumption}<!-- -->}<!-- -->})
\]
```

$$y = a + f(\underbrace{bx}_{\ge 0 \text{ by assumption}}) = a + f(\underbrace{bx}_{\mathclap{\ge 0 \text{ by assumption}}})$$

Figure 94

Alternatively, to use brackets instead of braces use `\underbracket` and `\overbracket` commands[6]:

```
\[
 z = \overbracket[3pt]{
     \underbracket{x}_{\text{real}<!---->} +
     \underbracket[0.5pt][7pt]{iy}_{\text{imaginary}<!---->}
     }^{\text{complex number}<!---->}
\]
```

$$z = \overbracket[3pt]{\underbracket{x}_{\text{real}} + \underbracket[0.5pt][7pt]{iy}_{\text{imaginary}}}^{\text{complex number}}$$

Figure 95

The optional arguments set the rule thickness and bracket height respectively:

```
\underbracket[rule thickness][bracket height]{argument}_{text below}
```

The `\xleftarrow` and `\xrightarrow` commands[7] produce arrows which extend to the length of the text. Yet again, the syntax is different: the optional argument (using `[` and `]`) specifies the subscript, and the mandatory argument (using `{` and `}`) specifies the superscript (which can be left empty by inserting a blank space).

6 requires the `mathtools` package

7 requires the `amsmath` package

```
\[
 A \xleftarrow{\text{this way}<!-- -->} B
  \xrightarrow[\text{or that way}]{ } C
\]
```

$$A \xleftarrow{\text{this way}} B \xrightarrow[\text{or that way}]{} C$$

For more extensible arrows, you must use `mathtools` package:

```
\[
 a \xleftrightarrow[under]{over} b\\
%
 A \xLeftarrow[under]{over} B\\
%
 B \xRightarrow[under]{over} C\\
%
 C \xLeftrightarrow[under]{over} D\\
%
 D \xhookleftarrow[under]{over} E\\
%
 E \xhookrightarrow[under]{over} F\\
%
 F \xmapsto[under]{over} G\\
\]
```

$$a \xleftrightarrow[under]{over} b$$
$$A \xLeftarrow[under]{over} B$$
$$B \xRightarrow[under]{over} C$$
$$C \xLeftrightarrow[under]{over} D$$
$$D \xhookleftarrow[under]{over} E$$
$$E \xhookrightarrow[under]{over} F$$
$$F \xmapsto[under]{over} G$$

Figure 96

and for harpoons:

```
\[
 H \xrightharpoondown[under]{over} I\\
%
 I \xrightharpoonup[under]{over} J\\
%
 J \xleftharpoondown[under]{over} K\\
%
 K \xleftharpoonup[under]{over} L\\
%
 L \xrightleftharpoons[under]{over} M\\
%
 M \xleftrightharpoons[under]{over} N
\]
```

$$H \xrightharpoondown[under]{over} I$$
$$I \xrightharpoonup[under]{over} J$$
$$J \xleftharpoondown[under]{over} K$$
$$K \xleftharpoonup[under]{over} L$$
$$L \xrightleftharpoons[under]{over} M$$
$$M \xleftrightharpoons[under]{over} N$$

Figure 97

28.2.2. `align` and `align*`

The `align` and `align*` environments, available through the `amsmath` package, are used for arranging equations of multiple lines. As with matrices and tables, `\\` specifies a line break, and `&` is used to indicate the point at which the lines should be aligned.

The `align*` environment is used like the `displaymath` or `equation*` environment:

```
\begin{align*}
 f(x) &= (x+a)(x+b) \\
 &= x^2 + (a+b)x + ab
\end{align*}
```

$$\begin{aligned} f(x) &= (x+a)(x+b) \\ &= x^2+(a+b)x+ab \end{aligned}$$

Note that the `align` environment must not be nested inside an `equation` (or similar) environment. Instead, `align` is a replacement for such environments; the contents inside an `align` are automatically placed in math mode.

`align*` suppresses numbering. To force numbering on a specific line, use the `\tag{...}` command before the linebreak.

`align` is similar, but automatically numbers each line like the `equation` environment. Individual lines may be referred to by placing a `\label{...}` before the linebreak. The `\nonumber` or `\notag` command can be used to suppress the number for a given line:

```
\begin{align}
 f(x) &= x^4 + 7x^3 + 2x^2 \nonumber \\
 &\qquad {} + 10x + 12
\end{align}
```

$$\begin{aligned} f(x) &= x^4+7x^3+2x^2 \\ &\qquad {}+10x+12 \qquad (3) \end{aligned}$$

Notice that we've added some indenting on the second line. Also, we need to insert the double braces (`{}`) before the + sign, otherwise latex won't create the correct spacing after the + sign. The reason for this is that without the braces, latex interprets the + sign as a unary operator, instead of the binary operator that it really is.

More complicated alignments are possible, with additional `&`'s on a single line specifying multiple "equation columns", each of which is aligned. The following example illustrates the alignment rule of `align*`:

```
\begin{align*}
 f(x)  &= a x^2+b x +c   &   g(x)  &= d x^3 \\
 f'(x) &= 2 a x +b       &   g'(x) &= 3 d x^2
\end{align*}
```

$$\begin{aligned} f(x) &= ax^2+bx+c & g(x) &= dx^3 \\ f'(x) &= 2ax+b & g'(x) &= 3dx^2 \end{aligned}$$

28.2.3. Braces spanning multiple lines

If you want a brace to continue across a new line, do the following:

```
\begin{align}
 f(x) &= \pi \left\{ x^4 + 7x^3 + 2x^2 \right.\nonumber\\
 &\qquad \left. {} + 10x + 12 \right\}
\end{align}
```

$$f(x) = \pi \Big\{ x^4 + 7x^3 + 2x^2$$
$$\qquad + 10x + 12 \} \qquad (4)$$

In this construction, the sizes of the left and right braces are not automatically equal, in spite of the use of `\left\{` and `\right\}`. This is because each line is typeset as a completely separate equation —notice the use of `\right.` and `\left.` so there are no unpaired `\left` and `\right` commands within a line (these aren't needed if the formula is on one line). You can control the size of the braces manually with the `\big`, `\Big`, `\bigg`, and `\Bigg` commands.

Alternatively, the height of the taller equation can be replicated in the other using the `\vphantom` command:

```
\begin{align}
 A &=     \left(\int_t XXX       \right.\nonumber\\
   &\qquad \left.\vphantom{\int_t} YYY \dots \right)
\end{align}
```

$$A = \left(\int_t XXX \right.$$
$$\qquad \left. \vphantom{\int_t} YYY \dots \right) \qquad (5)$$

Using aligned braces for piecewise functions

You can also use `\left\{` and `\right.` to typeset piecewise functions[8]:

```
\[f(x) = \left\{
  \begin{array}{lr}
    x^2 & : x < 0\\
    x^3 & : x \ge 0
  \end{array}
\right.
\]
```

$$f(x) = \left\{ \begin{array}{lr} x^2 & : x < 0 \\ x^3 & : x \ge 0 \end{array} \right.$$

8 http://en.wikipedia.org/wiki/piecewise%20functions

28.2.4. The cases environment

The `cases` environment[9] allows the writing of piecewise functions:

```
\[
 u(x) =
  \begin{cases}
   \exp{x} & \text{if } x \geq 0 \\
   1       & \text{if } x < 0
  \end{cases}
\]
```

$$u(x) = \begin{cases} \exp x & \text{if } x \geq 0 \\ 1 & \text{if } x < 0 \end{cases}$$

LaTeX will then take care of defining and or aligning the columns.

Within `cases`, text style math is used with results such as:

$$a = \begin{cases} \int x \, \mathrm{d} x \\ b^2 \end{cases}$$

Display style may be used instead, by using the `dcases` environment[10] from `mathtools`:

```
\[
 a =
   \begin{dcases}
     \int x\, \mathrm{d} x\\
     b^2
   \end{dcases}
\]
```

$$a = \begin{dcases} \int x \, \mathrm{d} x \\ b^2 \end{dcases}$$

Often the second column consists mostly of normal text, to set it in the normal roman font of the document the `dcases*` environment may be used:[11]

```
\[
 f(x) = \begin{dcases*}
        x  & when $x$ is even\\
        -x & when $x$ is odd
        \end{dcases*}
\]
```

9 requires the `amsmath` package
10 requires the `mathtools` package
11 requires the `mathtools` package

$$f(x) = \begin{cases} x & \text{when } x \text{ is even} \\ -x & \text{when } x \text{ is odd} \end{cases}$$

28.2.5. Other environments

Although `align` and `align*` are the most useful, there are several other environments which may also be of interest:

Environment name	Description	Notes
`eqnarray` and `eqnarray*`	Similar to `align` and `align*`	Not recommended because spacing is inconsistent
`multline` and `multline*`[12]	First line left aligned, last line right aligned	Equation number aligned vertically with first line and not centered as with other environments
`gather` and `gather*`[13]	Consecutive equations without alignment	
`flalign` and `flalign*`[14]	Similar to `align`, but left aligns first equation column, and right aligns last column	
`alignat` and `alignat*`[15]	Takes an argument specifying number of columns. Allows to control explicitly the horizontal space between equations	This environment takes one argument, the number of "equation columns": count the maximum number of `&`s in any row, add 1 and divide by 2. [ftp://ftp.ams.org/ams/doc/amsmath/amsldoc.pdf]

There are also few environments that don't form a math environment by themselves and can be used as building blocks for more elaborate structures:

Math environment name	Description
`gathered`[16]	Allows to gather few equations to be set under each other and assigned a single equation number
`split`[17]	Similar to `align*`, but used inside another displayed mathematics environment
`aligned`[18]	Similar to `align`, to be used inside another mathematics environment.
`alignedat`[19]	Similar to `alignat`, and just as it, takes an additional argument specifying number of columns of equations to set.

For example:

12 requires the `amsmath` package
13 requires the `amsmath` package
14 requires the `amsmath` package
15 requires the `amsmath` package
16 requires the `amsmath` package
17 requires the `amsmath` package
18 requires the `amsmath` package
19 requires the `amsmath` package

```
\begin{equation}
 \left.\begin{aligned}
        B'&=-\partial \times E,\\
        E'&=\partial \times B - 4\pi j,
       \end{aligned}
 \right\}
 \qquad \text{Maxwell's equations}
\end{equation}
```

$$\left.\begin{aligned} B'&=-\partial \times E,\\ E'&=\partial \times B - 4\pi j, \end{aligned} \right\} \qquad \text{Maxwell's equations} \tag{1.1}$$

```
\begin{alignat}{2}
 \sigma_1 &= x + y  &\quad \sigma_2 &= \frac{x}{y} \\
 \sigma_1' &= \frac{\partial x + y}{\partial x} & \sigma_2'
    &= \frac{\partial \frac{x}{y}<!---->}{\partial x}
\end{alignat}
```

$$\sigma_1 = x + y \qquad \sigma_2 = \frac{x}{y} \tag{1}$$

$$\sigma_1' = \frac{\partial x + y}{\partial x} \quad \sigma_2' = \frac{\partial \frac{x}{y}}{\partial x} \tag{2}$$

```
\begin{gather*}
a_0=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}f(x)\,\mathrm{d}x\\[6pt]
\begin{split}
a_n=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}f(x)\cos nx\,\mathrm{d}x=\\
=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}x^2\cos nx\,\mathrm{d}x
\end{split}\\[6pt]
\begin{split}
b_n=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}f(x)\sin nx\,\mathrm{d}x=\\
=\frac{1}{\pi}\int\limits_{-\pi}^{\pi}x^2\sin nx\,\mathrm{d}x
\end{split}\\[6pt]
\end{gather*}
```

$$a_0 = \frac{1}{\pi} \int\limits_{-\pi}^{\pi} f(x)\,\mathrm{d}x$$

$$\begin{aligned} a_n = \frac{1}{\pi} \int\limits_{-\pi}^{\pi} f(x) \cos nx \,\mathrm{d}x = \\ = \frac{1}{\pi} \int\limits_{-\pi}^{\pi} x^2 \cos nx \,\mathrm{d}x \end{aligned}$$

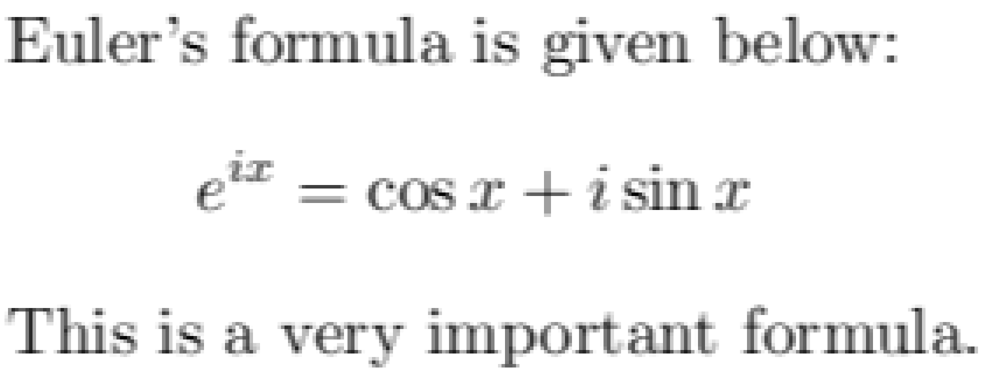
$$\begin{aligned} b_n = \frac{1}{\pi} \int\limits_{-\pi}^{\pi} f(x) \sin nx \,\mathrm{d}x = \\ = \frac{1}{\pi} \int\limits_{-\pi}^{\pi} x^2 \sin nx \,\mathrm{d}x \end{aligned}$$

Figure 98

28.3. Indented Equations

In order to indent an equation, you can set `fleqn` in the document class and then specify a certain value for the `\mathindent` variable:

```
\documentclass[a4paper,fleqn]{report}
\usepackage{amsmath}
\setlength{\mathindent}{1cm}
\begin{document}
\noindent Euler's formula is given below:
\begin{equation*}
 e^{ix} = \cos{x} + i \sin{x}.
\end{equation*}
\noindent This is a very important formula.
\end{document}
```


Euler's formula is given below:

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

This is a very important formula.

Figure 99

28.4. Page breaks in math environments

To suggest LaTeX insert a page break inside an `amsmath` environment you may use the `\displaybreak` command before the line break. Just like with `\pagebreak`, `\displaybreak` can take an optional argument between 0 and 4 denoting the level of desirability of a page break. While 0 means "it is permissible to break here", 4 forces a break. No argument means the same as 4.

Alternatively, you may enable automatic page breaks in math environments with `\allowdisplaybreaks`. It too can have an optional argument denoting the priority of page breaks in equations. Similarly, 1 means "allow page breaks but avoid them" and 4 means "break whenever you want". You can prohibit a page break after a given line using `\\*`.

LaTeX will insert a page break into a long equation if it has additional text added using `\intertext{}` without any additional commands.

Specific usage may look like this:

```
\begin{align*}
 &\vdots\\
 &=12+7 \int_0^2
  \left(
    -\frac{1}{4}\left(e^{-4t_1}+e^{4t_1-8}\right)
  \right)\,dt_1\displaybreak[3]\\
 &= 12-\frac{7}{4}\int_0^2 \left( e^{-4t_1}+e^{4t_1-8} \right)\,dt_1\\
 &\vdots %
\end{align*}
```

$$\vdots$$

$$= 12 + 7\int_0^2 \left(-\frac{1}{4}\left(e^{-4t_1} + e^{4t_1-8}\right)\right) dt_1$$

1

$$= 12 - \frac{7}{4}\int_0^2 e^{-4t_1} + e^{4t_1-8}\, dt_1$$

$$\vdots$$

Figure 100

28.5. Boxed Equations

For a single equation or alignment building block, with the tag outside the box, use `\boxed{}`:

```
\begin{equation}
 \boxed{x^2+y^2 = z^2}
\end{equation}
```

$$\boxed{x^2 + y^2 = z^2} \qquad (1)$$

Figure 101

If you want the entire line or several equations to be boxed, use a `minipage` inside an `\fbox{}`:

```
\fbox{
 \addtolength{\linewidth}{-2\fboxsep}%
 \addtolength{\linewidth}{-2\fboxrule}%
 \begin{minipage}{\linewidth}
  \begin{equation}
   x^2+y^2=z^2
  \end{equation}
 \end{minipage}
}
```

$$\boxed{\qquad\qquad x^2+y^2=z^2 \qquad\qquad (1)}$$

Figure 102

There is also the mathtools `\Aboxed{}` which is able to box across alignment marks:

```
\begin{align*}
\Aboxed{ f(x) & = \int h(x)\, dx} \\
              & = g(x)
\end{align*}
```

$$\begin{aligned} \boxed{f(x) = \int h(x)\, dx} \\ = g(x) \end{aligned}$$

Figure 103

28.6. Custom operators

Although many common operators[20] are available in LaTeX, sometimes you will need to write your own, e.g. to typeset the argmax[21] operator. The `\operatorname` and `\operatorname*` commands[22] display custom operators; the `*` version sets the underscored option underneath like the `\lim` operator:

20 Chapter 27.4 on page 308
21 `http://en.wikipedia.org/wiki/Arg%20max`
22 requires the `amsmath` package

```
\[
 \operatorname{arg\,max}_a f(a)
 = \operatorname*{arg\,max}_b f(b)
\]
```

$$\operatorname{arg\,max}_a f(a) = \operatorname{arg\,max}_b f(b)$$

However, if the operator is frequently used, it is preferable to define a new operator that can be used throughout the entire document. The `\DeclareMathOperator` and `\DeclareMathOperator*` commands[23] are specified in the header of the document:

```
\DeclareMathOperator*{\argmax}{arg\,max}
```

This defines a new command which may be referred to in the body:

```
\[
 \argmax_c f(c)
\]
```

$$\operatorname*{arg\,max}_c f(c)$$

28.7. Advanced formatting

28.7.1. Limits

There are defaults for placement of subscripts and superscripts e.g. limits for the `lim` operator are usually placed below the symbol:

```
\begin{equation}
  \lim_{a\to \infty} \tfrac{1}{a}
\end{equation}
```

$\lim_{a\to\infty} \tfrac{1}{a}$

To override this behavior, use the `\nolimits` operator:

23 requires the `amsmath` package

```
\begin{equation}
  \lim\nolimits_{a\to \infty} \tfrac{1}{a}
\end{equation}
```

$$\lim\nolimits_{a\to \infty} \tfrac{1}{a}$$

A `lim` in running text (inside `$...$`) will have its limits placed on the side, so that additional leading won't be required. To override this behavior, use the `\limits` command.

Similarly one can put subscripts under a symbol that usually have them on the side:

```
\begin{equation}
  \int_a^b x^2  \mathrm{d} x
\end{equation}
```

$$\textstyle\int_a^b x^2 \mathrm{d}x$$

Limits below and under:

```
\begin{equation}
  \int\limits_a^b x^2  \mathrm{d} x
\end{equation}
```

$$\textstyle\int\limits_a^b x^2 \mathrm{d}x$$

To change the default placement of summation-type symbols to the side for every case, add the `nosumlimits` option to the `amsmath` package. To change the placement for integral symbols, add `intlimits` to the options. `nonamelimits` can be used to change the default for named operators like `det` , `min` , `lim` , etc.

To produce one-sided limits, use `\underset` as follows:

```
\begin{equation}
  \lim_{a \underset{>}{\to} 0} \frac{1}{a}
\end{equation}
```

$$\lim\nolimits_{a \underset{>}{\to} 0} \tfrac{1}{a}$$

28.7.2. Subscripts and superscripts

While you can place symbols in subscript or superscript (in summation style symbols) with `\nolimits`:

```
\begin{equation}
  \sum\nolimits' C_n
\end{equation}
```

$\sum\nolimits' C_n$

It's impossible to mix them with typical usage of such symbols:

```
\begin{equation}
  \sum_{n=1}\nolimits' C_n
\end{equation}
```

$\sum_{n=1}\nolimits' C_n$

To add both prime and a limit to a symbol, one might use `\sideset` command:

```
\begin{equation}
  \sideset{}{'}\sum_{n=1}C_n
\end{equation}
```

$$\sideset{}{'}\sum_{n=1}C_n$$

It is very flexible: for example, to put letters in each corner of the symbol use this command:

```
\begin{equation}
  \sideset{_a^b}{_c^d}\sum
\end{equation}
```

$$\sideset{_a^b}{_c^d}\sum$$

If you wish to place them on the corners of an arbitrary symbol, you should use `\fourIdx` from the `fouridx` package.

However, a simple grouping can solve the problem:

```
\begin{equation}
  {\sum\limits_{n=1} }'C_n
\end{equation}
```

$${\sum\limits_{n=1}}'C_n$$

since a math operator can be used with limits or no limits. If you want to change its state, simply group it. You can make it another math operator if you want, and then you can have limits and then limits again.

28.7.3. Multiline subscripts

To produce multiline subscript use `\substack` command:

```
\begin{equation}
  \prod_{\substack{
            1\le i \le n\\
            1\le j \le m}<!---->}
     M_{i,j}
\end{equation}
```

$$\prod_{\substack{1\le i\le n\\ 1\le j\le m}} M_{i,j}$$

28.8. Text in aligned math display

To add small interjections in math environments use `\intertext` command:

```
\begin{minipage}{3in}
\begin{align*}
\intertext{If}
   A &= \sigma_1+\sigma_2\\
   B &= \rho_1+\rho_2\\
\intertext{then}
C(x) &= e^{Ax^2+\pi}+B
\end{align*}
\end{minipage}
```

If

$$A = \sigma_1 + \sigma_2$$
$$B = \rho_1 + \rho_2$$

then

$$C(x) = e^{Ax^2+\pi} + B$$

Figure 104

Note that any usage of this command does not change the alignment.

Also, in the above example, the command `\shortintertext{}` from the `mathtools` package could have been used instead of intertext to reduce the amount of vertical whitespace added between the lines.

28.9. Changing font size

There may be a time when you would prefer to have some control over the size. For example, using text-mode maths, by default a simple fraction will look like this: $\frac{a}{b}$, whereas you may prefer to have it displayed larger, like when in display mode, but still keeping it in-line, like this: $\displaystyle\frac{a}{b}$.

A simple approach is to utilize the predefined sizes for maths elements:

Size command	**Description**
`\displaystyle`	Size for equations in display mode
`\textstyle`	Size for equations in text mode
`\scriptstyle`	Size for first sub/superscripts
`\scriptscriptstyle`	Size for subsequent sub/superscripts

A classic example to see this in use is typesetting continued fractions (though it's better to use the `\cfrac` command[24] described in the Mathematics[25] chapter over the method provided below). The following code provides an example.

24 requires the `amsmath` package
25 Chapter 27.6.1 on page 312

```
\begin{equation}
  x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}<!---->}<!---->}
\end{equation}
```

$$x = a_0 + \frac{1}{a_1+\frac{1}{a_2+\frac{1}{a_3+a_4}}}$$

As you can see, as the fractions continue, they get smaller (although they will not get any smaller as in this example, they have reached the `\scriptstyle` limit). If you wanted to keep the size consistent, you could declare each fraction to use the display style instead, e.g.:

```
\begin{equation}
  x = a_0 + \frac{1}{\displaystyle a_1
          + \frac{1}{\displaystyle a_2
          + \frac{1}{\displaystyle a_3 + a_4}<!---->}<!---->}
\end{equation}
```

$$x = a_0 + \frac{1}{\displaystyle a_1 + \frac{1}{\displaystyle a_2 + \frac{1}{\displaystyle a_3 + a_4}}}$$

Another approach is to use the `\DeclareMathSizes` command to select your preferred sizes. You can only define sizes for `\displaystyle`, `\textstyle`, etc. One potential downside is that this command sets the global maths sizes, as it can only be used in the document preamble.

However, it's fairly easy to use: `\DeclareMathSizes{ds}{ts}{ss}{sss}`, where *ds* is the *display size* , *ts* is the *text size* , etc. The values you input are assumed to be point (pt) size.

NB the changes only take place if the value in the first argument matches the current document text size. It is therefore common to see a set of declarations in the preamble, in the event of the main font being changed. E.g.,

```
\DeclareMathSizes{10}{18}{12}{8}   % For size 10 text
\DeclareMathSizes{11}{19}{13}{9}   % For size 11 text
\DeclareMathSizes{12}{20}{14}{10}  % For size 12 text
```

28.10. Forcing \displaystyle for all math in a document

Put

```
\everymath{\displaystyle}
```

before

```
\begin{document}
```

to force all math to

```
\displaystyle
```

.

28.11. Adjusting vertical whitespace around displayed math

There are four parameters which control the vertical whitespace around displayed math:

```
\abovedisplayskip=12pt
\belowdisplayskip=12pt
\abovedisplayshortskip=0pt
\belowdisplayshortskip=7pt
```

Short skips are used if the preceding line ends, horizontally, before the formula. These parameters must be set after

```
\begin{document}
```

.

28.12. Notes

29. Theorems

With "theorem[1]" we can mean any kind of labelled enunciation that we want to look separated from the rest of the text and with sequential numbers next to it. This approach is commonly used for theorems in mathematics, but can be used for anything. LaTeX provides a command that will let you easily define any theorem-like enunciation.

29.1. Basic theorems

First of all, make sure you have the amsthm package enabled:

```
\usepackage{amsthm}
```

The easiest is the following:

```
\newtheorem{name}{Printed output}
```

put it in the preamble. The first argument is the name you will use to reference it, the second argument is the output LaTeX will print whenever you use it. For example:

```
\newtheorem{mydef}{Definition}
```

will define the `mydef` environment; if you use it like this:

```
\begin{mydef}
Here is a new definition
\end{mydef}
```

It will look like this:

Definition 3 *Here is a new definition*

with line breaks separating it from the rest of the text.

29.2. Theorem counters

Often the counters are determined by section, for example "Theorem 2.3" refers to the 3rd theorem in the 2nd section of a document. In this case, specify the theorem as follows:

```
\newtheorem{name}{Printed output}[numberby]
```

1 http://en.wikipedia.org/wiki/Theorem

where *numberby* is the name of the section level[2] (section/subsection/etc.) at which the numbering is to take place.

By default, each theorem uses its own counter. However it is common for similar types of theorems (e.g. Theorems, Lemmas and Corollaries) to share a counter. In this case, define subsequent theorems as:

```
\newtheorem{name}[counter]{Printed output}
```

where *counter* is the name of the counter to be used. Usually this will be the name of the master theorem.

The \newtheorem command may have at most one optional argument.

You can also create a theorem environment that is not numbered by using the `newtheorem*` command[3]. For instance,

```
\newtheorem*{mydef}{Definition}
```

defines the `mydef` environment, which will generate definitions without numbering. This requires `amsthm` package.

29.3. Proofs

The `proof` environment[4] can be used for adding the proof of a theorem. The basic usage is:

```
\begin{proof}
Here is my proof
\end{proof}
```

It just adds *Proof* in italics at the beginning of the text given as argument and a white square (Q.E.D.[5] symbol, also known as a tombstone[6]) at the end of it. If you are writing in another language than English, just use babel[7] with the right argument and the word *Proof* printed in the output will be translated accordingly; anyway, in the source the name of the environment remains `proof` .

If you would like to manually name the proof, include the name in square brackets:

```
\begin{proof}[Proof of important theorem]
Here is my important proof
\end{proof}
```

If the last line of the proof is displayed math then the Q.E.D. symbol will appear on a subsequent empty line. To put the Q.E.D. symbol at the end of the last line, use the `\qedhere` command:

2 Chapter 5.3.3 on page 56
3 Requires the `amsthm` package
4 Requires the `amsthm` package
5 `http://en.wikipedia.org/wiki/Q.E.D.`
6 `http://en.wikipedia.org/wiki/Tombstone%20%28typography%29`
7 Chapter 12 on page 133

```
\begin{proof}
Here is my proof:
\[
a^2 + b^2 = c^2 \qedhere
\]
\end{proof}
```

The method above does not work with the deprecated environment `eqnarray*` . Use `align*` instead.

To use a custom Q.E.D. symbol, redefine the `\qedsymbol` command. To hide the Q.E.D. symbol altogether, redefine it to be blank:

```
\renewcommand{\qedsymbol}{}
```

29.4. Theorem styles

It adds the possibility to change the output of the environments defined by `\newtheorem` using the `\theoremstyle` command[8] in the header:

```
\theoremstyle{stylename}
```

the argument is the style you want to use. All subsequently defined theorems will use this style. Here is a list of the possible pre-defined styles:

`stylename`	Description	**Appearance**
`plain`	Used for theorems, lemmas, propositions, etc. (default)	**Theorem 1.** *Theorem text.*
`definition`	Used for definitions and examples	**Definition 2.** Definition text.
`remark`	Used for remarks and notes	*Remark* 3. Remark text.

29.4.1. Custom styles

To define your own style, use the `\newtheoremstyle` command[9]:

```
\newtheoremstyle{stylename}% name of the style to be used
  {spaceabove}% measure of space to leave above the theorem. E.g.: 3pt
  {spacebelow}% measure of space to leave below the theorem. E.g.: 3pt
  {bodyfont}% name of font to use in the body of the theorem
  {indent}% measure of space to indent
  {headfont}% name of head font
  {headpunctuation}% punctuation between head and body
  {headspace}% space after theorem head; " " = normal interword space
  {headspec}% Manually specify head
```

(Any arguments that are left blank will assume their default value). Here is an example *headspec* :

8 Requires the `amsthm` package
9 Requires the `amsthm` package

```
\thmname{#1}\thmnumber{ #2}:\thmnote{ #3}
```

which would look something like:

Definition 2 : Topology

for the following:

```
\begin{definition}[Topology]...
```

(The note argument, which in this case is Topology, is always optional, but will not appear by default unless you specify it as above in the head spec).

29.5. Conflicts

The theorem environment conflicts with other environments, for example *wrapfigure* . A work around is to redefine theorem, for example the following way:

```
% Fix latex
\def\smallskip{\vskip\smallskipamount}
\def\medskip{\vskip\medskipamount}
\def\bigskip{\vskip\bigskipamount}

% Hand made theorem
\newcounter{thm}[section]
\renewcommand{\thethm}{\thesection.\arabic{thm}}
\def\claim#1{\par\medskip\noindent\refstepcounter{thm}\hbox{\bf
 \arabic{chapter}.\arabic{section}.\arabic{thm}. #1.}
\it\ %\ignorespaces
}
\def\endclaim{
\par\medskip}
\newenvironment{thm}{\claim}{\endclaim}
```

In this case theorem looks like:

```
\begin{thm}{Claim}\label{lyt-prob}
Let it be.
Then you know.
\end{thm}
```

29.6. Notes

29.7. External links

- [ftp://ftp.ams.org/pub/tex/doc/amscls/amsthdoc.pdf `amsthm` documentation]

30. Chemical Graphics

chemfig[1] is a package used to draw 2D chemical structures. It is an alternative to ochem[2]. Whereas ochem requires Perl to draw chemical structures, chemfig uses the tikz[3] package to produce its graphics. chemfig is used by adding the following to the preamble:

```
\usepackage{chemfig}
```

30.1. Basic Usage

The primary command used in this package is `\chemfig{}` :

```
\chemfig{<atom1><bond type>[<angle>,<coeff>,<tikz code>]<atom2>}
```

<angle> is the bond angle between two atoms (or nodes). There are three types of angles: absolute, relative, and predefined. Absolute angles give a precise angle (generally, 0 to 360, though they can also be negative), and are represented with the syntax `[:<absolute angle>]` . Relative angles require the syntax `[::<relative angle>]` and produce an angle relative to the angle of the preceding bond. Finally, predefined angles are whole numbers from 0 to 7 indicating intervals of 45 degrees. These are produced with the syntax `[< predefined angle>]` . The predefined angles and their corresponding absolute angles are represented in the diagram below.

```
\chemfig{(-[:0,1.5,,,draw=none]\scriptstyle\color{red}0)
(-[1]1)(-[:45,1.5,,,draw=none]\scriptstyle\color{red}45)
(-[2]2)(-[:90,1.5,,,draw=none]\scriptstyle\color{red}90)
(-[3]3)(-[:135,1.5,,,draw=none]\scriptstyle\color{red}135)
(-[4]4)(-[:180,1.5,,,draw=none]\scriptstyle\color{red}180)
(-[5]5)(-[:225,1.5,,,draw=none]\scriptstyle\color{red}225)
(-[6]6)(-[:270,1.5,,,draw=none]\scriptstyle\color{red}270)
(-[7]7)(-[:315,1.5,,,draw=none]\scriptstyle\color{red}315)
-0}
```

1 http://www.ctan.org/tex-archive/macros/latex/contrib/chemfig/
2 http://www.2k-software.de/ingo/ochem.html
3 http://az.ctan.org/pkg/pgf

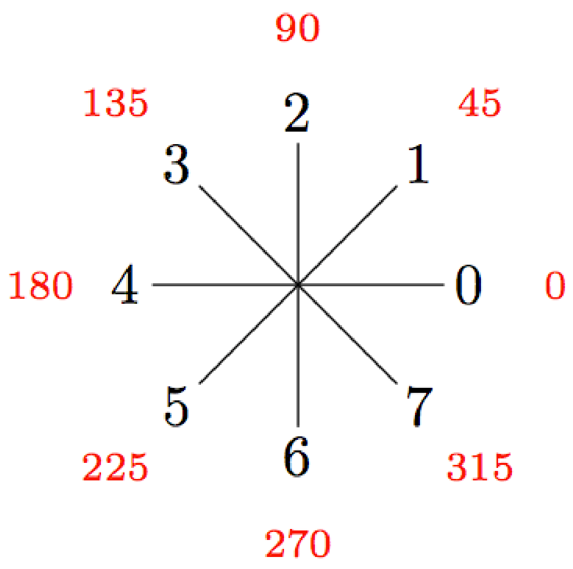


Figure 105

<bond type> describes the bond attaching <atom1> and <atom2>. There are 9 different bond types:

```
\chemfig{A-B}\\
\chemfig{A=B}\\
\chemfig{A~B}\\
\chemfig{A>B}\\
\chemfig{A<B}\\
\chemfig{A>:B}\\
\chemfig{A<:B}\\
\chemfig{A>B}\\
\chemfig{A<B}\\
```

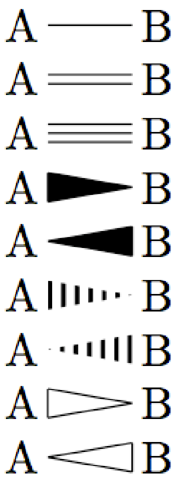


Figure 106

<coeff> represents the factor by which the bond's length will be multiplied.

<tikz code> includes additional options regarding the color or style of the bond.

A methane molecule, for instance, can be produced with the following code:

```
\chemfig{C(-[:0]H)(-[:90]H)(-[:180]H)(-[:270]H)}
```

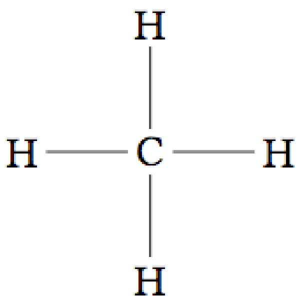


Figure 107

Linear molecules (such as methane) are a weak example of this, but molecules are formed in chemfig by nesting.

30.2. Skeletal Diagrams

Skeleton diagrams can be produced as follows:

```
\chemfig{-[:30]-[:-30]-[:30]}
```

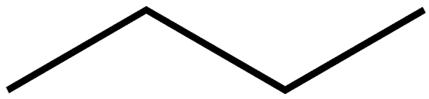


Figure 108

```
\chemfig{-[:30]=[:-30]-[:30]}
```

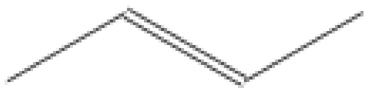


Figure 109

30.3. Rings

Rings follow the syntax `<atom>*<n>(code)` , where ”n” indicates the number of sides in the ring and ”code” represents the specific content of each ring (bonds and atoms).

```
\chemfig{A*6(-B-C-D-E-F-)}
```

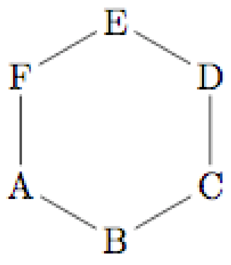


Figure 110

```
\chemfig{A*5(-B-C-D-E-)}
```

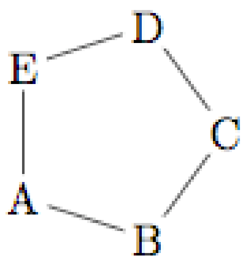


Figure 111

```
\chemfig{*6(=-=-=-)}
```

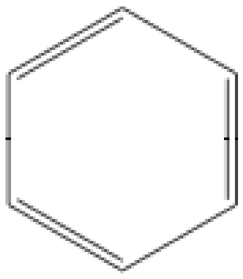


Figure 112

```
\chemfig{**5(------)}
```

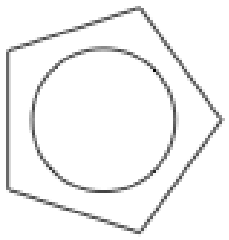


Figure 113

30.4. Lewis Structures

Lewis structures use the syntax \lewis{<n1><n2>...<ni>,<atom>}, where <ni> is a number between 0 and 7 representing the position of the electrons. By default, the electrons are represented by a dash (-). Appending a period (.) or colon (:) after a number will display single and paired electrons respectively.

```
\lewis{0.2.4.6.,C}
```

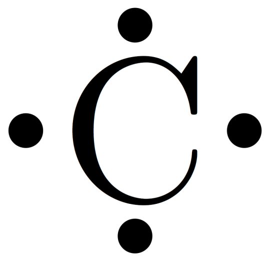


Figure 114

Lewis structures can also be included within \chemfig{}.

```
\chemfig{H-[:52.24]\lewis{1:3:,O}-[::-104.48]H}
```

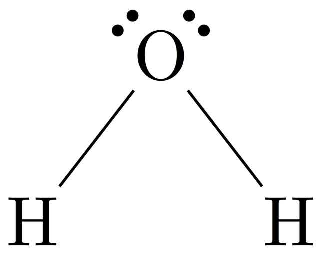


Figure 115

30.5. Ions

For example, consider an acetate ion:

```
\chemfig{-(-[1]O^{-})=[7]O}
```

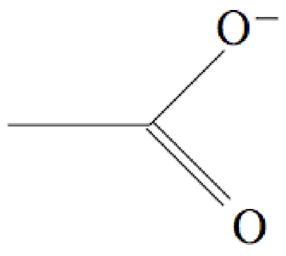


Figure 116

Because the chemfig commands enters the math mode, ion charges can be added as superscripts (one caveat: a negative ion requires that the minus sign be enclosed in brackets, as in the example).

The charge of an ion can be circled by using `\oplus` and `\ominus` :

```
\chemfig{-(-[1]O^{\ominus})=[7]O}
```

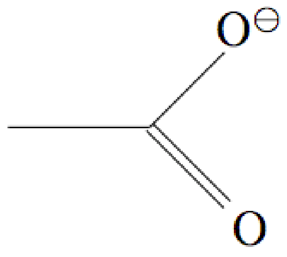


Figure 117

Alternatively, charges can be placed above ions using \chemabove{}{}:

```
\chemfig{-\chemabove{N}{\scriptstyle\oplus}(=[1]O)-[7]O^{\ominus}}
```

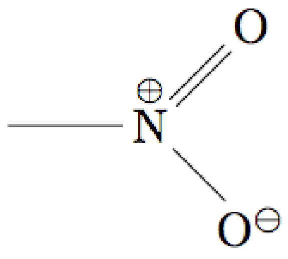


Figure 118

30.6. Resonance Structures and Formal Charges

Resonance structures require a few math commands:

```
% see "Advanced Mathematics" for use of \left and \right
% add to preamble:
%        \usepackage{mathtools}        % \Longleftrightarrow
$\left\{\chemfig{O-N(=[:60]O)-[:300]O}\right\}
\Longleftrightarrow
\left\{\chemfig{O=N(-[:60]O)-[:300]O}\right\}
\Longleftrightarrow
\left\{\chemfig{O-N(-[:60]O)=[:300]O}\right\}$
```

30.7. Chemical Reactions

Chemical reactions can be created with the following commands:

```
\chemrel[<arg1>][<arg2>]{<arrow code>}
```

```
\chemsign+        % produces a +
```

In `\chemrel{}` , <arg1> and <arg2> represent text placed above and below the arrow, respectively.

There are four types of arrows that can be produced with `\chemrel{}` :

```
A\chemrel{->}B\par
A\chemrel{<-}B\par
A\chemrel{<->}B\par
A\chemrel{<>}B
```

30.8. Naming Chemical Graphics

Molecules can be named with the command

```
\chemname[<dim>]{\chemfig{<code of the molecule>}}{<name>}
```

<dim> is inserted between the bottom of the molecule and the top of the name defined by <name>. It is 1.5ex by default.

<name> will be centered relative to the molecule it describes.

```
\chemname{\chemfig{R-C(-[:-30]OH)=[:30]O}}{Carboxylic acid}
\chemsign{+}
\chemname{\chemfig{R'OH}}{Alcohol}
\chemrel{->}
\chemname{\chemfig{R-C(-[:-30]OR')=[:30]O}}{Ester}
\chemsign{+}
\chemname{\chemfig{H_2O}}{Water}
```

In the reaction above, `\chemname{}` inserts 1.5ex plus the depth of the carboxylic acid molecule in between each molecule and their respective names. This is because the graphic for the first molecule in the reaction (carboxylic acid) extends deeper than the rest of the molecules. A different result is produced by putting the alcohol first:

```
\chemname{\chemfig{R'OH}}{Alcohol}
\chemsign{+}
\chemname{\chemfig{R-C(-[:-30]OH)=[:30]O}}{Carboxylic acid}
\chemrel{->}
\chemname{\chemfig{R-C(-[:-30]OR')=[:30]O}}{Ester}
\chemsign{+}
\chemname{\chemfig{H_2O}}{Water}
```

This is fixed by adding `\chemnameinit{<deepest molecule>}` before the first instance of `\chemname{}` in a reaction and by adding `\chemnameinit{}` after the reaction:

```
\chemnameinit{\chemfig{R-C(-[:-30]OH)=[:30]O}}
\chemname{\chemfig{R'OH}}{Alcohol}
\chemsign{+}
\chemname{\chemfig{R-C(-[:-30]OH)=[:30]O}}{Carboxylic acid}
\chemrel{->}
\chemname{\chemfig{R-C(-[:-30]OR')=[:30]O}}{Ester}
\chemsign{+}
\chemname{\chemfig{H_2O}}{Water}
\chemnameinit{}
```

Lastly, adding \\ in <name> will produce a line-break, allowing the name to span multiple lines.

30.9. Advanced Graphics

For advanced commands and examples, refer to the chemfig manual[4], where a more thorough and complete introduction to the package can be found.

30.10. mhchem Package

mhchem[5] is a package used to typeset chemical formulae and equations. As well as typeset basic 2D chemical structures. To use this package, add the following to your preamble:

```
\usepackage[version=3]{mhchem}
```

Chemical species are included using the `\ce` command. For example

```
\ce{3H2O} \\
\ce{1/2H2O} \\
\ce{AgCl2-} \\
\ce{H2_{(aq)}} \\
```

A few things here are automatically typeset; The 2 in `\ce{H2O}` is automatically subscripted without requiring additional commands. The amount of the species precedes the formula. 1/2 and other fractional amounts are automatically typeset as in `\ce{1/2H2O}` . The

4 `http://mirror.ctan.org/macros/latex/contrib/chemfig/chemfig_doc_en.pdf`
5 `http://www.ctan.org/tex-archive/macros/latex/contrib/mhchem/`

charge in `\ce{AgCl2-}` is automatically superscripted. If the charge is neither 1 or -1, a ^ will superscript it, as in `\ce{AgCl2-}` . The phase is not automatically subscripted and needs to be enclosed in parenthesis preceded with a _ as in `\ce{H2_{(aq)}` .

30.11. XyMTeX package

The following code produces the image for corticosterone[6] below.

```
\documentclass{letter}
\usepackage{epic,carom}
\pagestyle{empty}
\begin{document}
\begin{picture}(1000,500)
   \put(0,0){\ste
roid[d]{3D==O;{{10}}==\lmoiety{H$_{3}$C};{{13}}==\lmoiety{H$_{3}$C};{{11}}==HO}}
   \put(684,606){\sixunitv{}{2D==O;1==OH}{cdef}}
\end{picture}
\end{document}
```

6 http://en.wikipedia.org/wiki/Corticosterone

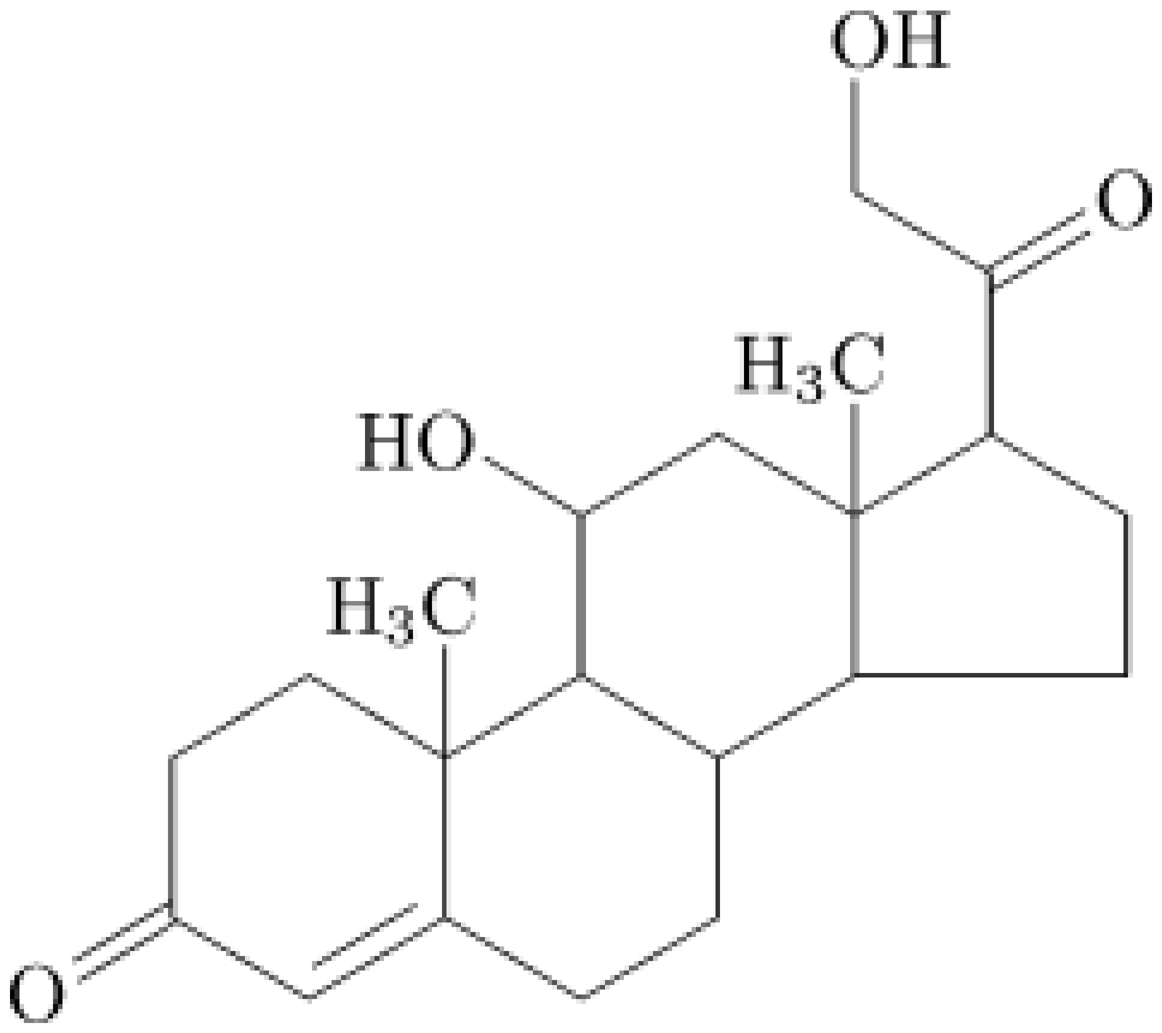


Figure 119 Corticosterone as rendered by XyMTeX

31. Algorithms

LaTeX has several packages for typesetting algorithms in form of "pseudocode[1]". They provide stylistic enhancements over a uniform style (i.e., all in typewriter font) so that constructs such as loops or conditionals are visually separated from other text. For typesetting *real* code, written in a *real* programming language, consider the *listings* package described in Source Code Listings[2].

31.1. Typesetting using the `algorithmic` package

The `algorithmic` package uses a different set of commands than the `algorithmicx` package. This is not compatible with `revtex4-1` . Basic commands are:

```
\STATE <text>
\IF{<condition>} \STATE{<text>} \ELSE \STATE{<text>} \ENDIF
\FOR{<condition>} \STATE{<text>} \ENDFOR
\FOR{<condition> \TO <condition> } \STATE{<text>} \ENDFOR
\FORALL{<condition>} \STATE{<text>} \ENDFOR
\WHILE{<condition>} \STATE{<text>} \ENDWHILE
\REPEAT \STATE{<text>} \UNTIL{<condition>}
\LOOP \STATE{<text>} \ENDLOOP
\REQUIRE <text>
\ENSURE <text>
\RETURN <text>
\PRINT <text>
\COMMENT{<text>}
\AND, \OR, \XOR, \NOT, \TO, \TRUE, \FALSE
```

Complete documentation is listed at `http://mirror.ctan.org/tex-archive/macros/latex/contrib/algorithms/algorithms.pdf`. Most commands are similar to the `algorithmicx` equivalents, but with different capitalization. The package `algorithms bundle` at the ctan repository[3], dated 2009-08-24, describes both the `algorithmic` environment (for typesetting algorithms) and the `algorithm` floating wrapper (see below[4]) which is designed to wrap around the algorithmic environment.

The `algorithmic` package is suggested for IEEE journals[5] as it is a part of their default style sheet.[6]

1 `http://en.wikipedia.org/wiki/pseudocode`
2 Chapter 32 on page 393
3 `http://mirror.ctan.org/tex-archive/macros/latex/contrib/algorithms/`
4 Chapter 31.5 on page 389
5 `http://ieeexplore.ieee.org/xpl/periodicals.jsp`
6 `http://www.ctan.org/tex-archive/macros/latex/contrib/IEEEtran`

31.2. Typesetting using the `algorithm2e` package

The `algorithm2e` package (first released 1995, latest updated January 2013 according to the v5.0 manual[7]) allows typesetting algorithms with a lot of customization. Like `algorithmic` , this package is also not compatible with Revtex-4.1.[8]

Unlike `algorithmic` , `algorithm2e` provides a relatively huge number of customization options to the algorithm suiting to the needs of various users. The CTAN-manual[9] provides a comprehensible list of examples and full set of controls.

Typically, the usage between `\begin{algorithm}` and `\end{algorithm}` would be

1. Declaring a set of keywords(to typeset as functions/operators), layout controls, caption, title, header text (which appears before the algorithm's main steps e.g.: Input,Output)

2. Writing the main steps of the algorithm, with each step ending with a \;

This may be taken in analogy with writing a latex-preamble before we start the actual document.

The package is loaded like

```
\usepackage[]{algorithm2e}
```

and a simple example, taken from the v4.01 manual, is

```
\begin{algorithm}[H]
 \KwData{this text}
 \KwResult{how to write algorithm with \LaTeX2e }
 initialization\;
 \While{not at end of this document}{
  read current\;
  \eIf{understand}{
   go to next section\;
   current section becomes this one\;
   }{
   go back to the beginning of current section\;
  }
 }
 \caption{How to write algorithms}
\end{algorithm}
```

which produces

7 http://mirror.ctan.org/tex-archive/macros/latex/contrib/algorithm2e/doc/algorithm2e.pdf
8 http://tex.stackexchange.com/questions/70181/revtex4-1-and-algorithm2e-indentation-clash
9 http://mirror.ctan.org/tex-archive/macros/latex/contrib/algorithm2e/doc/algorithm2e.pdf

```
Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while not at end of this document do
    read current;
    if understand then
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end
```

Algorithm 1: How to write algorithms

Figure 120

More details are in the manual hosted on the ctan website[10].

31.3. Typesetting using the `algorithmicx` package

The `algorithmicx` package provides a number of popular constructs for algorithm designs. Put `\usepackage{algpseudocode}` in the preamble to use the algorithmic environment to write algorithm pseudocode (`\begin{algorithmic}...\end{algorithmic}`). You might want to use the algorithm environment (`\usepackage{algorithm}`) to wrap your algorithmic code in an algorithm environment (`\begin{algorithm}...\end{algorithm}`) to produce a floating environment with numbered algorithms.

The command `\begin{algorithmic}` can be given the optional argument of a positive integer, which if given will cause line numbering to occur at multiples of that integer. E.g. `\begin{algorithmic}[5]` will enter the algorithmic environment and number every fifth line.

Below is an example of typesetting a basic algorithm using the `algorithmicx` package (remember to add the `\usepackage{algpseudocode}` statement to your document preamble):

```
\begin{algorithmic}
\If {$i\geq maxval$}
    \State $i\gets 0$
\Else
    \If {$i+k\leq maxval$}
        \State $i\gets i+k$
    \EndIf
\EndIf
\end{algorithmic}
```

10 http://mirror.ctan.org/tex-archive/macros/latex/contrib/algorithm2e/doc/algorithm2e.pdf

The LaTeX source can be written to a format familiar to programmers so that it is easy to read. This will not, however, affect the final layout in the document.

```
if $i \geq maxval$ then
    $i \gets 0$
else
    if $i + k \leq maxval$ then
        $i \gets i + k$
    end if
end if
```

Figure 121

Basic commands have the following syntax:

Statement (\State causes a new line, can also be used in front of other commands)

```
\State $x\gets <value>$
```

Three forms of if-statements:

```
\If{<condition>} <text> \EndIf
```

```
\If{<condition>} <text> \Else <text> \EndIf
```

```
\If{<condition>} <text> \ElsIf{<condition>} <text> \Else <text> \EndIf
```

The third form accepts as many `\ElsIf{}` clauses as required. Note that it is `\ElsIf` and not `\ElseIf` .

Loops:

```
\For{<condition>} <text> \EndFor
```

```
\ForAll{<condition>} <text> \EndFor
```

```
\While{<condition>} <text> \EndWhile
```

```
\Repeat <text> \Until{<condition>}
```

```
\Loop <text> \EndLoop
```

Pre- and postcondition:

```
\Require <text>
```

```
\Ensure <text>
```

Functions

```
\Function{<name>}{<params>} <body> \EndFunction
```

```
\Return <text>
```

```
\Call{<name>}{<params>}
```

This command will usually be used in conjunction with a `\State` command as follows:

```
\Function{Increment}{$a$}
    \State $a \gets a+1$
    \State \Return $a$
\EndFunction
```

Comments:

```
\Comment{<text>}
```

Note to users who switched from the old `algorithmic` package: comments may be placed everywhere in the source; there are no limitations as in the old `algorithmic` package.

31.3.1. Renaming things: algorithm to procedure, require/ensure to input/output

```
\floatname{algorithm}{Procedure}
\renewcommand{\algorithmicrequire}{\textbf{Input:}}
\renewcommand{\algorithmicensure}{\textbf{Output:}}
```

31.3.2. Custom algorithmic blocks

The `algorithmicx` package allows you to define your own environments.

To define blocks beginning with a starting command and ending with an ending command, use

```
\algblock[<block>]{<start>}{<end>}
```

This defines two commands `\<start>` and `\<end>` which have no parameters. The text displayed by them is `\textbf{<start>}` and `\textbf{<end>}` .

With `\algblockdefx` you can give the text to be output by the starting and ending command and the number of parameters for these commands. In the text the n-th parameter is referenced by `#n` .

```
\algblockdefx[<block>]{<start>}{<end>}
    [<startparamcount>][<default value>]{<start text>}
    [<endparamcount>][<default value>]{<end text>}
```

Example:

```
\algblock[Name]{Start}{End}
\algblockdefx[NAME]{START}{END}%
    [2][Unknown]{Start #1(#2)}%
    {Ending}
\algblockdefx[NAME]{}{OTHEREND}%
    [1]{Until (#1)}
\begin{algorithmic}
\Start
    \Start
        \START[One]{x}
        \END
        \START{0}
        \OTHEREND{\texttt{True}}
    \End
    \Start
    \End
\End
\end{algorithmic}
```

More advanced customization and other constructions are described in the `algorithmicx` manual: `http://mirror.ctan.org/macros/latex/contrib/algorithmicx/algorithmicx.pdf`

31.4. The `algorithm` environment

It is often useful for the algorithm produced by `algorithmic` to be "floated" to the optimal point in the document to avoid it being split across pages. The `algorithm` environment provides this and a few other useful features. Include it by adding the

`\usepackage{algorithm}` to your document's preamble. It is entered into by

```
\begin{algorithm}
\caption{<your caption for this algorithm>}
\label{<your label for references later in your document>}
\begin{algorithmic}
<algorithmic environment>
\end{algorithmic}
\end{algorithm}
```

31.4.1. Algorithm numbering

The default numbering system for the `algorithm` package is to number algorithms sequentially. This is often not desirable, particularly in large documents where numbering according to chapter is more appropriate. The numbering of algorithms can be influenced by providing the name of the document component within which numbering should be recommenced. The legal values for this option are: part, chapter, section, subsection, subsubsection or nothing (default). For example:

```
\usepackage[chapter]{algorithm}
```

31.4.2. List of algorithms

When you use figures or tables, you can add a list of them close to the table of contents; the `algorithm` package provides a similar command. Just put

```
\listofalgorithms
```

anywhere in the document, and LaTeX will print a list of the "algorithm" environments in the document with the corresponding page and the caption.

31.4.3. An example from the manual

This is an example taken from the manual (official manual, p.14[11])

```
\begin{algorithm}                    % enter the algorithm environment
\caption{Calculate $y = x^n$}        % give the algorithm a caption
\label{alg1}                         % and a label for \ref{} commands later
 in the document
\begin{algorithmic}                  % enter the algorithmic environment
    \REQUIRE $n \geq 0 \vee x \neq 0$
    \ENSURE $y = x^n$
    \STATE $y \Leftarrow 1$
    \IF{$n < 0$}
        \STATE $X \Leftarrow 1 / x$
        \STATE $N \Leftarrow -n$
    \ELSE
        \STATE $X \Leftarrow x$
        \STATE $N \Leftarrow n$
    \ENDIF
    \WHILE{$N \neq 0$}
        \IF{$N$ is even}
            \STATE $X \Leftarrow X \times X$
            \STATE $N \Leftarrow N / 2$
        \ELSE[$N$ is odd]
            \STATE $y \Leftarrow y \times X$
            \STATE $N \Leftarrow N - 1$
        \ENDIF
    \ENDWHILE
\end{algorithmic}
\end{algorithm}
```

More information about all possible commands available at the project page

`http://developer.berlios.de/docman/?group_id=3442`

The official manual is located at

`http://mirrors.ctan.org/macros/latex/contrib/algorithms/algorithms.pdf`

31.5. Typesetting using the `program` package

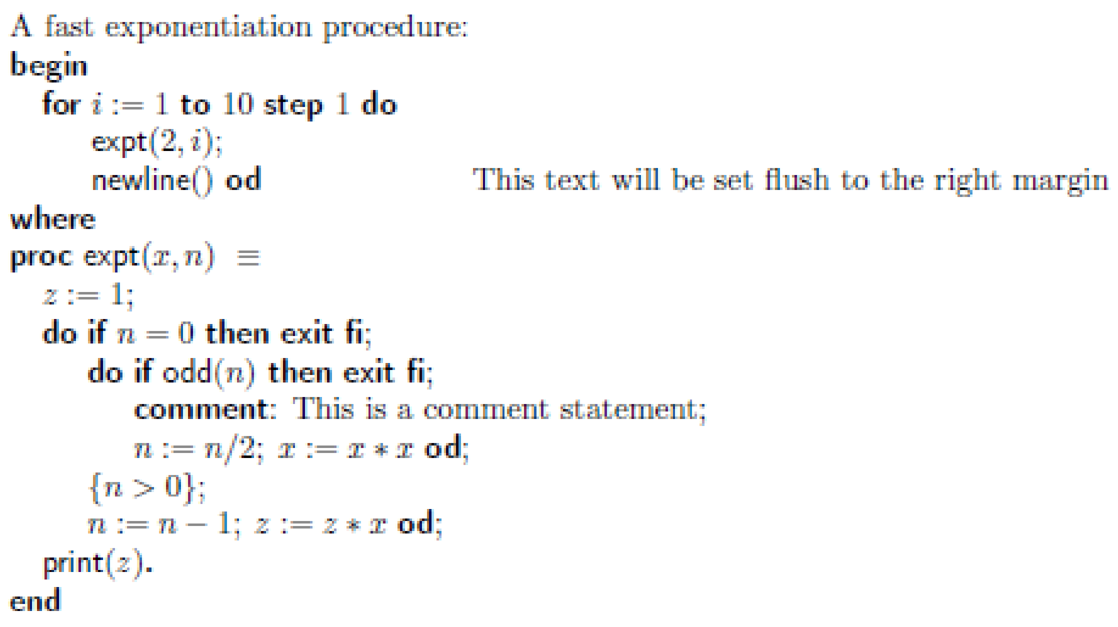
The `program` package provides macros for typesetting algorithms. Each line is set in math mode, so all the indentation and spacing is done automatically. The notation `|variable_name|` can be used within normal text, maths expressions or programs to indicate a

11 Chapter 31.5 on page 389

variable name. Use `\origbar` to get a normal | symbol in a program. The commands `\A` , `\B` , `\P` , `\Q` , `\R` , `\S` , `\T` and `\Z` typeset the corresponding bold letter with the next object as a subscript (eg `\S1` typesets `{\bf S$_1$}` etc). Primes work normally, eg `\S‘‘` .

Below is an example of typesetting a basic algorithm using the `program` package (remember to add the `\usepackage{program}` statement to your document preamble):

```
\begin{program}
\mbox{A fast exponentiation procedure:}
\BEGIN \\ %
  \FOR i:=1 \TO 10 \STEP 1 \DO
     |expt|(2,i); \\ |newline|() \OD %
\rcomment{This text will be set flush to the right margin}
\WHERE
\PROC |expt|(x,n) \BODY
          z:=1;
          \DO \IF n=0 \THEN \EXIT \FI;
             \DO \IF |odd|(n) \THEN \EXIT \FI;
\COMMENT{This is a comment statement};
                n:=n/2; x:=x*x \OD;
             \{ n>0 \};
             n:=n-1; z:=z*x \OD;
          |print|(z) \ENDPROC
\END
\end{program}
```



```
A fast exponentiation procedure:
begin
   for $i := 1$ to 10 step 1 do
       expt$(2, i)$;
       newline() od                    This text will be set flush to the right margin
where
proc expt$(x, n)$ $\equiv$
   $z := 1$;
   do if $n = 0$ then exit fi;
       do if odd$(n)$ then exit fi;
           comment: This is a comment statement;
           $n := n/2$; $x := x * x$ od;
       $\{n > 0\}$;
       $n := n - 1$; $z := z * x$ od;
   print$(z)$.
end
```

Figure 122

The commands `\(` and `\)` are redefined to typeset an algorithm in a minipage, so an algorithm can appear as a single box in a formula. For example, to state that a particular action system is equivalent to a WHILE loop you can write:

```
\[
\( \ACTIONS A:
        A \EQ \IF \B{} \THEN \S{}; \CALL A
                       \ELSE \CALL Z \FI \QE
```

```
   \ENDACTIONS \)
\EQT
\( \WHILE \B{} \DO \S{} \OD \)
\]
```

Dijkstra conditionals and loops:

```
\begin{program}
\IF x = 1 \AR y:=y+1
\BAR x = 2 \AR y:=y^2
\utdots
\BAR x = n \AR y:=\displaystyle\sum_{i=1}^n y_i \FI

\DO 2 \origbar x \AND x>0 \AR x:= x/2
\BAR \NOT 2 \origbar x    \AR x:= \modbar{x+3} \OD
\end{program}
```

Loops with multiple exits:

```
\begin{program}
\DO \DO \IF \B1 \THEN \EXIT \FI;
        \S1;
        \IF \B2 \THEN \EXIT(2) \FI \OD;
    \IF \B1 \THEN \EXIT \FI \OD
\end{program}
```

A Reverse Engineering Example.

Here's the original program:

```
\begin{program}
 \VAR \seq{m := 0, p := 0, |last| := `` ''};
 \ACTIONS |prog|:
|prog| \ACTIONEQ %
    \seq{|line| := `` '', m := 0, i := 1};
    \CALL |inhere| \ENDACTION
l \ACTIONEQ %
    i := i+1;
    \IF (i=(n+1)) \THEN \CALL |alldone| \FI ;
    m := 1;
    \IF |item|[i] \neq |last|
        \THEN |write|(|line|); |line| := `` ''; m := 0;
              \CALL |inhere| \FI ;
    \CALL |more| \ENDACTION
|inhere| \ACTIONEQ %
    p := |number|[i]; |line| := |item|[i];
    |line| := |line| \concat `` '' \concat p;
    \CALL |more| \ENDACTION
|more| \ACTIONEQ %
    \IF (m=1) \THEN p := |number|[i];
    |line| := |line| \concat ``, '' \concat p \FI ;
    |last| := |item|[i];
    \CALL l  \ENDACTION
|alldone| \ACTIONEQ |write|(|line|); \CALL Z \ENDACTION \ENDACTIONS \END
\end{program}
```

And here's the transformed and corrected version:

```
\begin{program}
\seq{|line| := `` '', i := 1};
\WHILE i \neq n+1 \DO
  |line| := |item|[i] \concat `` '' \concat |number|[i];
  i := i+1;
  \WHILE i \neq n+1 \AND |item|[i] = |item|[i-1] \DO
```

```
    |line| := |line| \concat ``, '' \concat |number|[i]);
    i := i+1 \OD ;
  |write|(|line|) \OD
\end{program}
```

The package also provides a macro for typesetting a set like this: `\set{x \in N | x > 0}` .

Lines can be numbered by setting `\NumberProgramstrue` and numbering turned off with `\NumberProgramsfalse`

Package page[12]

Package documentation[13]

31.6. References

- The official manual for the `algorithms` package, Rogério Brito (2009), `http://mirrors.ctan.org/macros/latex/contrib/algorithms/algorithms.pdf`

12 `http://www.ctan.org/pkg/program`

13 `http://mirror.ctan.org/macros/latex/contrib/program/program-doc.pdf`

32. Source Code Listings

32.1. Using the *listings* package

Using the package `listings` you can add non-formatted text as you would do with `\begin{verbatim}` but its main aim is to include the source code of any programming language within your document. If you wish to include pseudocode or algorithms, you may find Algorithms and Pseudocode[1] useful also.

To use the package, you need:

```
\usepackage{listings}
```

The `listings` package supports highlighting of all the most common languages and it is highly customizable. If you just want to write code within your document the package provides the `lstlisting` environment:

```
\begin{lstlisting}
Put your code here.
\end{lstlisting}
```

Another possibility, that is very useful if you created a program on several files and you are still editing it, is to import the code from the source itself. This way, if you modify the source, you just have to recompile the LaTeX code and your document will be updated. The command is:

```
\lstinputlisting{source_filename.py}
```

in the example there is a Python source, but it doesn't matter: you can include any file but you have to write the full file name. It will be considered plain text and it will be highlighted according to your settings, that means it doesn't recognize the programming language by itself. You can specify the language while including the file with the following command:

```
\lstinputlisting[language=Python]{source_filename.py}
```

You can also specify a scope for the file.

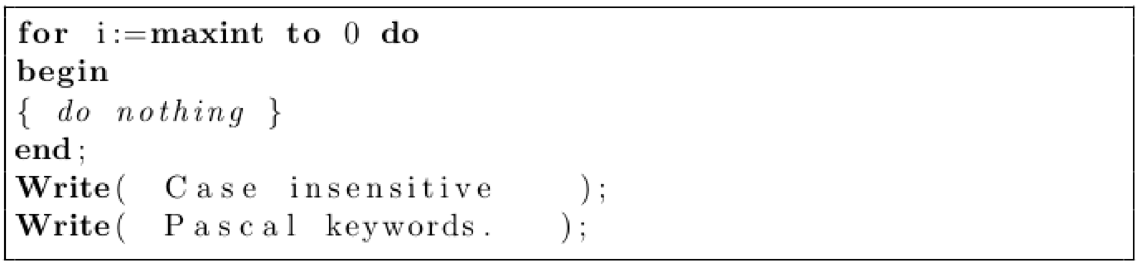
```
\lstinputlisting[language=Python, firstline=37, lastline=45]{source_filename.py}
```

This comes in handy if you are sure that the file will not change (at least before the specified lines). You may also omit the `firstline` or `lastline` parameter: it means *everything up to or starting from this point* .

1 http://en.wikibooks.org/wiki/LaTeX%2FAlgorithms%20and%20Pseudocode

This is a basic example for some Pascal code:

```
\documentclass{article}
\usepackage{listings}             % Include the listings-package
\begin{document}
\lstset{language=Pascal}          % Set your language (you can change the
 language for each code-block optionally)

\begin{lstlisting}[frame=single]  % Start your code-block
for i:=maxint to 0 do
begin
{ do nothing }
end;
Write('Case insensitive ');
Write('Pascal keywords.');
\end{lstlisting}

\end{document}
```



```
for i:=maxint to 0 do
begin
{ do nothing }
end;
Write( Case insensitive  );
Write( Pascal keywords.  );
```

Figure 123

32.1.1. Supported languages

It supports the following programming languages:

ABAP[2,4]
ACSL
Ada[4]
Algol[4]
Ant
Assembler[2,4]
Awk[4]
bash
Basic[2,4]
C[4]
C++[4]
Caml[4]
Clean
Cobol[4]
Comal
csh
Delphi
Eiffel
IDL[4]
inform
Java[4]
JVMIS
ksh
Lisp[4]
Logo
make[4]
Mathematica[1,4]
Matlab
Mercury
MetaPost
Miranda
Mizar
ML
Modelica[3]
Modula-2
MuPAD
PL/I
Plasm
POV
Prolog
Promela
Python
R
Reduce
Rexx
RSL
Ruby
S[4]
SAS
Scilab
sh
SHELXL
Simula[4]
SQL

Elan	NASTRAN	tcl[4]
erlang	Oberon-2	TeX[4]
Euphoria	OCL[4]	VBScript
Fortran[4]	Octave	Verilog
GCL	Oz	VHDL[4]
Gnuplot	Pascal[4]	VRML[4]
Haskell	Perl	XML
HTML	PHP	XSLT

For some of them, several dialects are supported. For more information, refer to the documentation that comes with the package, it should be within your distribution under the name `listings-*.dvi` .

Notes

1. It supports Mathematica code only if you are typing in plain text format. You can't include *.NB files `\lstinputlisting{...}` as you could with any other programming language, but Mathematica can export in a pretty-formatted LaTeX source.
2. Specification of the dialect is mandatory for these languages (e.g. `language={[x86masm]Assembler}`).
3. Modelica is supported via the dtsyntax package available here[2].
4. For these languages, multiple dialects are supported. C, for example, has ANSI, Handel, Objective and Sharp. See p. 12 of the [ftp://ftp.tex.ac.uk/tex-archive/macros/latex/contrib/listings/listings.pdf listings manual] for an overview.

32.1.2. Settings

You can modify several parameters that will affect how the code is shown. You can put the following code anywhere in the document (it doesn't matter whether before or after `\begin{document}`), change it according to your needs. The meaning is explained next to any line.

```
\usepackage{listings}
\usepackage{color}

\definecolor{mygreen}{rgb}{0,0.6,0}
\definecolor{mygray}{rgb}{0.5,0.5,0.5}
\definecolor{mymauve}{rgb}{0.58,0,0.82}

\lstset{ %
  backgroundcolor=\color{white},   % choose the background color; you must add
 \usepackage{color} or \usepackage{xcolor}
  basicstyle=\footnotesize,        % the size of the fonts that are used for the
 code
  breakatwhitespace=false,         % sets if automatic breaks should only happen
 at whitespace
  breaklines=true,                 % sets automatic line breaking
  captionpos=b,                    % sets the caption-position to bottom
  commentstyle=\color{mygreen},    % comment style
  deletekeywords={...},            % if you want to delete keywords from the
 given language
```

2 `https://code.google.com/p/dtsyntax/`

```
  escapeinside={\%*}{*)},          % if you want to add LaTeX within your code
  extendedchars=true,              % lets you use non-ASCII characters; for
 8-bits encodings only, does not work with UTF-8
  frame=single,                           % adds a frame around the code
  keepspaces=true,                 % keeps spaces in text, useful for keeping
 indentation of code (possibly needs columns=flexible)
  keywordstyle=\color{blue},       % keyword style
  language=Octave,                 % the language of the code
  otherkeywords={*,...},            % if you want to add more keywords to the
 set
  numbers=left,                    % where to put the line-numbers; possible
 values are (none, left, right)
  numbersep=5pt,                   % how far the line-numbers are from the code
  numberstyle=\tiny\color{mygray}, % the style that is used for the line-numbers
  rulecolor=\color{black},         % if not set, the frame-color may be changed
 on line-breaks within not-black text (e.g. comments (green here))
  showspaces=false,                % show spaces everywhere adding particular
 underscores; it overrides 'showstringspaces'
  showstringspaces=false,          % underline spaces within strings only
  showtabs=false,                  % show tabs within strings adding particular
 underscores
  stepnumber=2,                    % the step between two line-numbers. If it's
 1, each line will be numbered
  stringstyle=\color{mymauve},     % string literal style
  tabsize=2,                           % sets default tabsize to 2 spaces
  title=\lstname                   % show the filename of files included with
 \lstinputlisting; also try caption instead of title
}
```

escapeinside

The `escapeinside` line needs an explanation. The option `escapeinside={A}{B}` will define delimiters for escaping into LaTeX code, *i.e.* all the code between the string "A" and "B" will be parsed as LaTeX over the current *listings* style. In the example above, the comments for *Octave* start with `%`, and they are going to be printed in the document unless they start with `%*`, in which case they are read as LaTeX (with all LaTeX commands fulfilled) until they're closed with another `*)`. If you add the above paragraph, the following can be used to alter the settings within the code:

```
\lstset{language=C,caption={Descriptive Caption Text},label=DescriptiveLabel}
```

There are many more options, check the official documentation.

32.1.3. Style definition

The package lets you define styles, *i.e.* profiles specifying a set of settings.

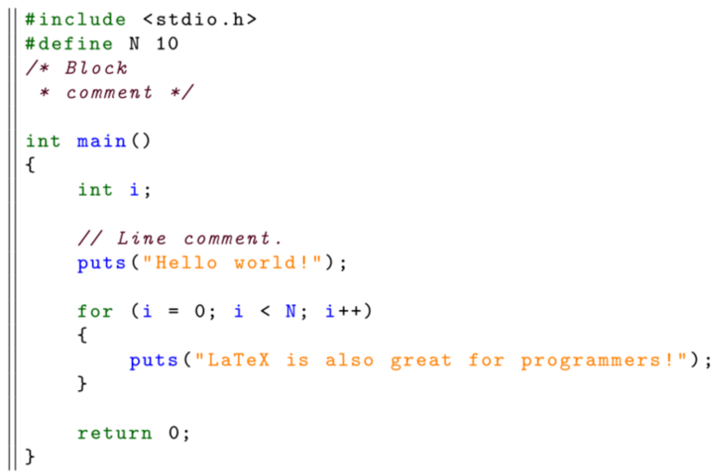
Example

```
\lstdefinestyle{customc}{
  belowcaptionskip=1\baselineskip,
  breaklines=true,
  frame=L,
  xleftmargin=\parindent,
  language=C,
  showstringspaces=false,
  basicstyle=\footnotesize\ttfamily,
  keywordstyle=\bfseries\color{green!40!black},
  commentstyle=\itshape\color{purple!40!black},
  identifierstyle=\color{blue},
  stringstyle=\color{orange},
}

\lstdefinestyle{customasm}{
  belowcaptionskip=1\baselineskip,
  frame=L,
  xleftmargin=\parindent,
  language=[x86masm]Assembler,
  basicstyle=\footnotesize\ttfamily,
  commentstyle=\itshape\color{purple!40!black},
}

\lstset{escapechar=@,style=customc}
```

In our example, we only set two options globally: the default style and the escape character. Usage:

```
\begin{lstlisting}
#include <stdio.h>
#define N 10
/* Block
 * comment */

int main()
{
    int i;

    // Line comment.
    puts("Hello world!");

    for (i = 0; i < N; i++)
    {
        puts("LaTeX is also great for programmers!");
    }

    return 0;
}
\end{lstlisting}

\lstinputlisting[caption=Scheduler, style=customc]{hello.c}
```

The C part will print as



```
#include <stdio.h>
#define N 10
/* Block
 * comment */

int main()
{
    int i;

    // Line comment.
    puts("Hello world!");

    for (i = 0; i < N; i++)
    {
        puts("LaTeX is also great for programmers!");
    }

    return 0;
}
```

Figure 124

32.1.4. Automating file inclusion

If you have a bunch of source files you want to include, you may find yourself doing the same thing over and over again. This is where macros show their real power.

```
\newcommand{\includecode}[2][c]{\lstinputlisting[caption=#2, escapechar=,
 style=custom#1]{#2}}
% ...

\includecode{sched.c}
\includecode[asm]{sched.s}
% ...

\lstlistoflistings
```

In this example, we create one command to ease source code inclusion. We set the default style to be *customc* . All listings will have their name as caption: we do not have to write the file name twice thanks to the macro. Finally we list all listings with this command from the `listings` package.

See Macros[3] for more details.

32.1.5. Encoding issue

By default, `listings` does not support multi-byte encoding for source code. The `extendedchar` option only works for 8-bits encodings such as latin1.

To handle UTF-8, you should tell listings how to interpret the special characters by defining them like so

```
\lstset{literate=
  {á}{{\'a}}1 {é}{{\'e}}1 {í}{{\'i}}1 {ó}{{\'o}}1 {ú}{{\'u}}1
  {Á}{{\'A}}1 {É}{{\'E}}1 {Í}{{\'I}}1 {Ó}{{\'O}}1 {Ú}{{\'U}}1
  {à}{{\`a}}1 {è}{{\`e}}1 {ì}{{\`i}}1 {ò}{{\`o}}1 {ù}{{\`u}}1
  {À}{{\`A}}1 {È}{{\'E}}1 {Ì}{{\`I}}1 {Ò}{{\`O}}1 {Ù}{{\`U}}1
  {ä}{{\"a}}1 {ë}{{\"e}}1 {ï}{{\"i}}1 {ö}{{\"o}}1 {ü}{{\"u}}1
  {Ä}{{\"A}}1 {Ë}{{\"E}}1 {Ï}{{\"I}}1 {Ö}{{\"O}}1 {Ü}{{\"U}}1
  {â}{{\^a}}1 {ê}{{\^e}}1 {î}{{\^i}}1 {ô}{{\^o}}1 {û}{{\^u}}1
  {Â}{{\^A}}1 {Ê}{{\^E}}1 {Î}{{\^I}}1 {Ô}{{\^O}}1 {Û}{{\^U}}1
  {œ}{{\oe}}1 {Œ}{{\OE}}1 {æ}{{\ae}}1 {Æ}{{\AE}}1 {ß}{{\ss}}1
  {ű}}1 {Ű}}1 {ő}}1 {Ő}}1
  {ç}{{\c c}}1 {Ç}{{\c C}}1 {ø}{{\o}}1 {å}{{\r a}}1 {Å}{{\r A}}1
  {€}{{\EUR}}1 {£}{{\pounds}}1
}
```

The above table will cover most characters in latin languages. For a more detailed explanation of the usage of the `literate` option check section 5.4 in the [ftp://ftp.tex.ac.uk/tex-archive/macros/latex/contrib/listings/listings.pdf Listings Documentation].

Another possibility is to replace `\usepackage{listings}` (in the preamble) with `\usepackage{listingsutf8}`.

3 Chapter 51 on page 569

32.1.6. Customizing captions

You can have fancy captions (or titles) for your listings using the `caption` package. Here is an example for `listings`.

```
\usepackage{caption}
\usepackage{listings}

\DeclareCaptionFont{white}{ \color{white} }
\DeclareCaptionFormat{listing}{
  \colorbox[cmyk]{0.43, 0.35, 0.35,0.01 }{
    \parbox{\textwidth}{\hspace{15pt}#1#2#3}
  }
}
\captionsetup[lstlisting]{ format=listing, labelfont=white, textfont=white,
 singlelinecheck=false, margin=0pt, font={bf,footnotesize} }

% ...

\lstinputlisting[caption=My caption]{sourcefile.lang}
```

32.2. The *minted* package

`minted` is an alternative to `listings` which has become popular. It uses the external Python library Pygments[4] for code highlighting, which as of Nov 2014 boasts over 300 supported languages and text formats.

As the package relies on external Python code, the setup require a few more steps than a usual LaTeX package, so please have a look at their GitHub repo[5] and their manual[6].

32.3. References

A lot more detailed information can be found in a PDF by Carsten Heinz and Brooks Moses[7].

Details and documentation about the Listings package can be found at its CTAN website[8].

4 `http://pygments.org/`
5 `https://github.com/gpoore/minted`
6 `https://github.com/gpoore/minted/blob/master/source/minted.pdf`
7 `http://mirror.hmc.edu/ctan/macros/latex/contrib/listings/listings.pdf`
8 `http://www.ctan.org/tex-archive/macros/latex/contrib/listings/`

33. Linguistics

There are a number of LaTeX packages available for writing linguistics papers. Various packages have been created for enumerated examples, syntactic trees, OT tableaux, feature matrices, IPA fonts, and many other applications. Some packages such as the

```
tipa
```

package are effectively standard within the field, while others will vary by author preference.

Some recommended packages[1]:

- Glosses: gb4e or Covington;
- IPA symbols: tipa;
- OT Tableaux: OTtablx;
- Syntactic trees: qtree + tree-dvips (for drawing arrows);

  - Alternatively, xyling is very powerful but not as user friendly as qtree;
  - The xy[2] package itself has a steep learning curve, but allows a lot of control; for simplest trees use the xymatrix feature and arrows;
  - tikz-qtree[3] has the same syntax as qtree, but uses PGF/TikZ, which allows more options for drawing arrows, etc.

- Dependency trees and bubble parses:

  - The TikZ-dependency[4] package provides a high-level, convenient interface to draw dependency graphs. It is based on PGF/TikZ but does not require prior knowledge of TikZ in order to be used productively.

- Attribute-Value Matrices (AVMs): avm[5]
- John Frampton's expex: expex[6]

33.1. Enumerated examples

There are several commonly used packages for creating the kinds of numbered examples that are used in linguistics publications.

1 `http://jones.ling.indiana.edu/~mdickinson/08/latex/slides.pdf` LaTeX for Linguists presentation
2 `http://ctan.org/tex-archive/macros/generic/diagrams/xypic/xy`
3 `http://ctan.org/pkg/tikz-qtree`
4 `http://sourceforge.net/projects/tikz-dependency/`
5 `http://nlp.stanford.edu/~manning/tex/avm.sty`
6 `http://www.math.neu.edu/ling/tex/`

33.1.1. gb4e

The

`gb4e`

package[7] is called with:

```
\usepackage{gb4e}
```

IMPORTANT: If you use gb4e package, let it be **the last \usepackage call** in the document's preamble. Otherwise you may get exceeded parameter stack size error.

Examples for this package are placed within the

`exe`

environment, and each example is introduced with the `\ex` command.

```
\begin{exe}
        \ex This is an example.
\end{exe}
```

produces:

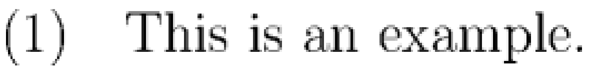


Figure 125

Multiple examples can be included within the environment, and each will have its own number.

```
\begin{exe}
        \ex This is the first example.
        \ex This is the second example.
        \ex This is the third.
\end{exe}
```

produces:

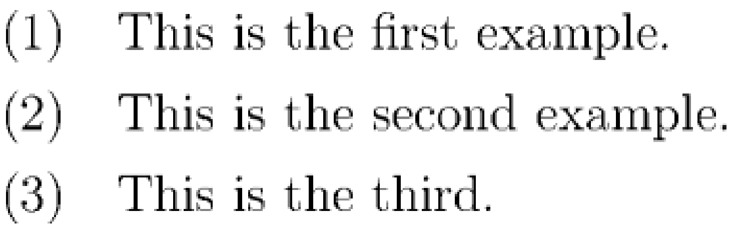


Figure 126

To create nested lists of examples, the

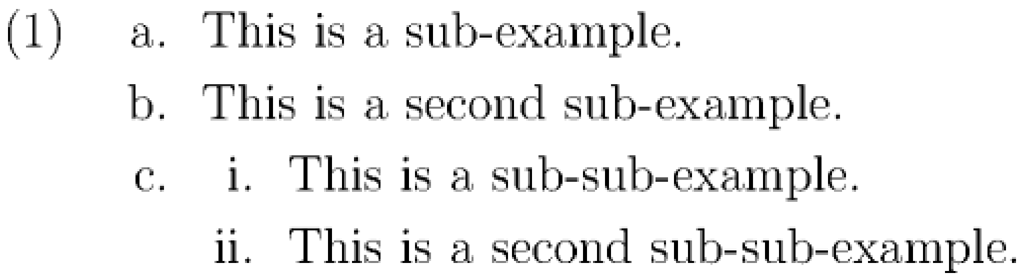
7 `http://ctan.mines-albi.fr/help/Catalogue/entries/gb4e.html/` The gb4e package on CTAN

```
xlist
```

enviroment is used.

```
\begin{exe}
    \ex \begin{xlist}
        \ex This is a sub-example.
        \ex This is a second sub-example.
        \ex \begin{xlist}
            \ex This is a sub-sub-example.
            \ex This is a second sub-sub-example.
        \end{xlist}
    \end{xlist}
\end{exe}
```

produces:



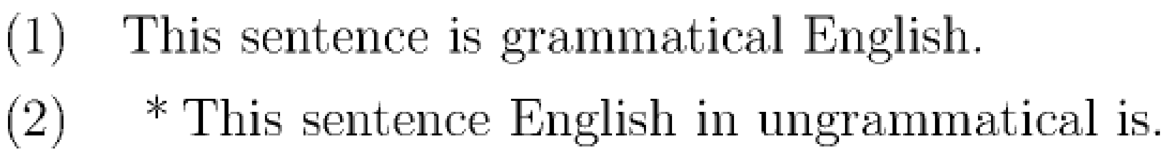
(1) a. This is a sub-example.
b. This is a second sub-example.
c. i. This is a sub-sub-example.
ii. This is a second sub-sub-example.

Figure 127

For notating acceptability judgments, the `\ex` command can take an optional argument. When including a judgment marker, the corresponding sentence must be surrounded by braces.

```
\begin{exe}
        \ex This sentence is grammatical English.
        \ex[*] {This sentence English in ungrammatical is.}
\end{exe}
```

produces:



(1) This sentence is grammatical English.
(2) * This sentence English in ungrammatical is.

Figure 128

Referencing examples in text works as it does in normal LaTeX documents. See the labeling and cross-referencing[8] section for more details.

```
\begin{exe}
        \ex\label{ex1} Godzilla destroyed the city.
        \ex\label{ex2} Godzilla roared.
```

8 Chapter 21 on page 267

```
\end{exe}
Sentence (\ref{ex1}) contains two arguments, but (\ref{ex2}) contains only one.
```

Further details can be found in the full documentation available here[9].

33.1.2. lingmacros

The

```
lingmacros
```

package[10] created by Emma Pease is an alternate method for example numbering. This package uses two main commands, `\enumsentence` and `\eenumsentence`. The former is used for singleton examples, while the latter command is used for nested examples.

```
\enumsentence{This is an example.}
```

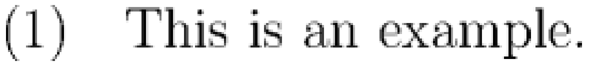


Figure 129

```
\enumsentence{This is the first example.}
\enumsentence{This is the second example.}
\enumsentence{This is the third.}
```

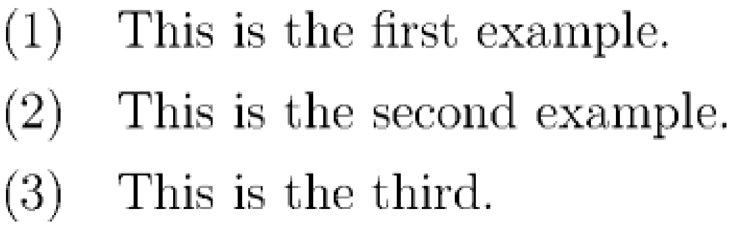


Figure 130

Multiply nested examples make use of the normal LaTeX list environments[11].

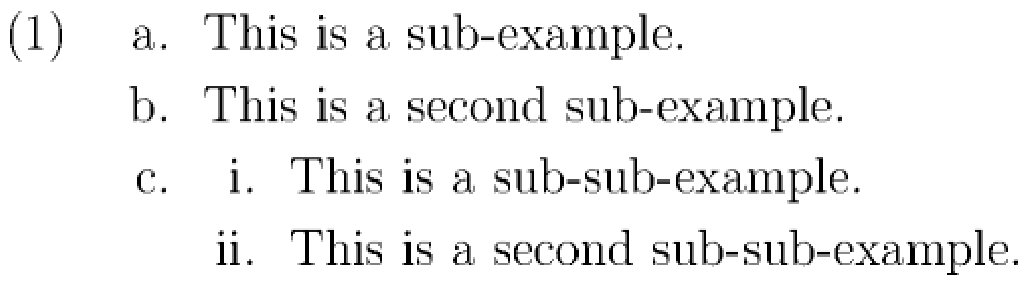
```
\eenumsentence{\item This is a sub-example.
               \item This is a second sub-example.
               \item \begin{enumerate}
                      \item This is sub-sub-example.
                      \item This is a second sub-sub-example.
                      \end{enumerate}
                }
```

produces:

9 http://ctan.mackichan.com/macros/latex/contrib/gb4e/gb4e-doc.pdf

10 http://ctan.org/tex-archive/macros/latex209/contrib/trees/tree-dvips The lingmacros package on CTAN

11 Chapter 10 on page 109



(1) a. This is a sub-example.
b. This is a second sub-example.
c. i. This is a sub-sub-example.
ii. This is a second sub-sub-example.

Figure 131

Full documentation can be found here[12].

33.2. Syntactic trees

Often, linguists will have to illustrate the syntactic structure of a sentence. One device for doing this is through syntactic trees. Unfortunately, trees look very different in different grammar formalisms, and different LaTeX packages are suited for different formalisms.

33.2.1. Constituent trees

While there are several packages for drawing syntactic trees available for LaTeX, this article focuses on the qtree and xyling packages.

qtree

Drawing trees with qtree is relatively straightforward. First, the `qtree` package has to be included in the document's preamble:

```
\usepackage{qtree}
```

A new tree is started using the **Tree** command, each (sub-)tree is indicated by brackets **[ ]** . The root of a (sub-)tree is always preceded by a **.** , leaf nodes are simply expressed by their labels.

For example, the following code

```
\Tree [.S [.NP LaTeX ] [.VP [.V is ] [.NP fun ] ] ]
```

produces this syntactic tree as output:

12 http://mirrors.ibiblio.org/pub/mirrors/CTAN/macros/latex209/contrib/trees/tree-dvips/lingmacros-manual.pdf

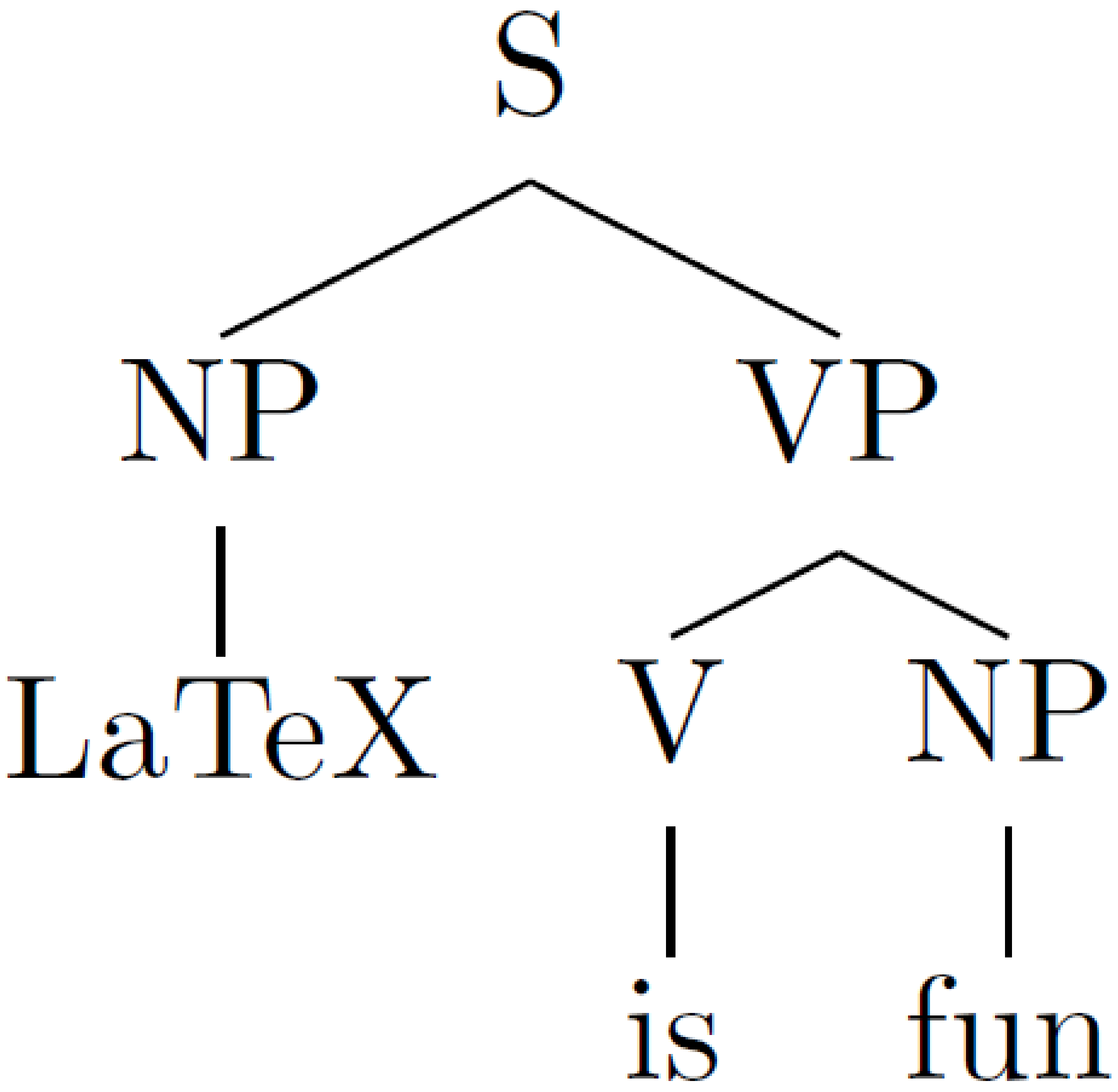


Figure 132

Note that the spaces before the closing brackets are **mandatory** .

By default, qtree centers syntactic trees on the page. This behaviour can be turned off by either specifying the behaviour when loading the package

```
\usepackage[nocenter]{qtree} % do not center trees
```

or via the command

```
\qtreecenterfalse % do not center trees from here on
```

anywhere in the document. The effect of the latter can be undone by using the command

```
\qtreecentertrue % center trees from here on
```

IMPORTANT: If you use gb4e package, let it be the last **usepackage** call in the document's preamble. Otherwise you may get exceeded parameter stack size error.

tikz-qtree

Using the same syntax as qtree, tikz-qtree is another easy-to-use alternative for drawing syntactic trees.

For simple trees, tikz-qtree is completely interchangable with qtree. However, some of qtree's advanced features are implemented in a different way, or not at all. On the other hand, tikz-qtree provides other features such as controlling the direction of the tree's growth (top to bottom, left to right etc.) or different styles for edges.

To use the `tikz-qtree` package for drawing trees, put the following into the document's preamble:

```
\usepackage{tikz}
\usepackage{tikz-qtree}
```

The syntax of `tikz-qtree` and result when drawing a simple tree is the same as for `qtree`.

```
\Tree [.S [.NP LaTeX ] [.VP [.V is ] [.NP fun ] ] ]
```

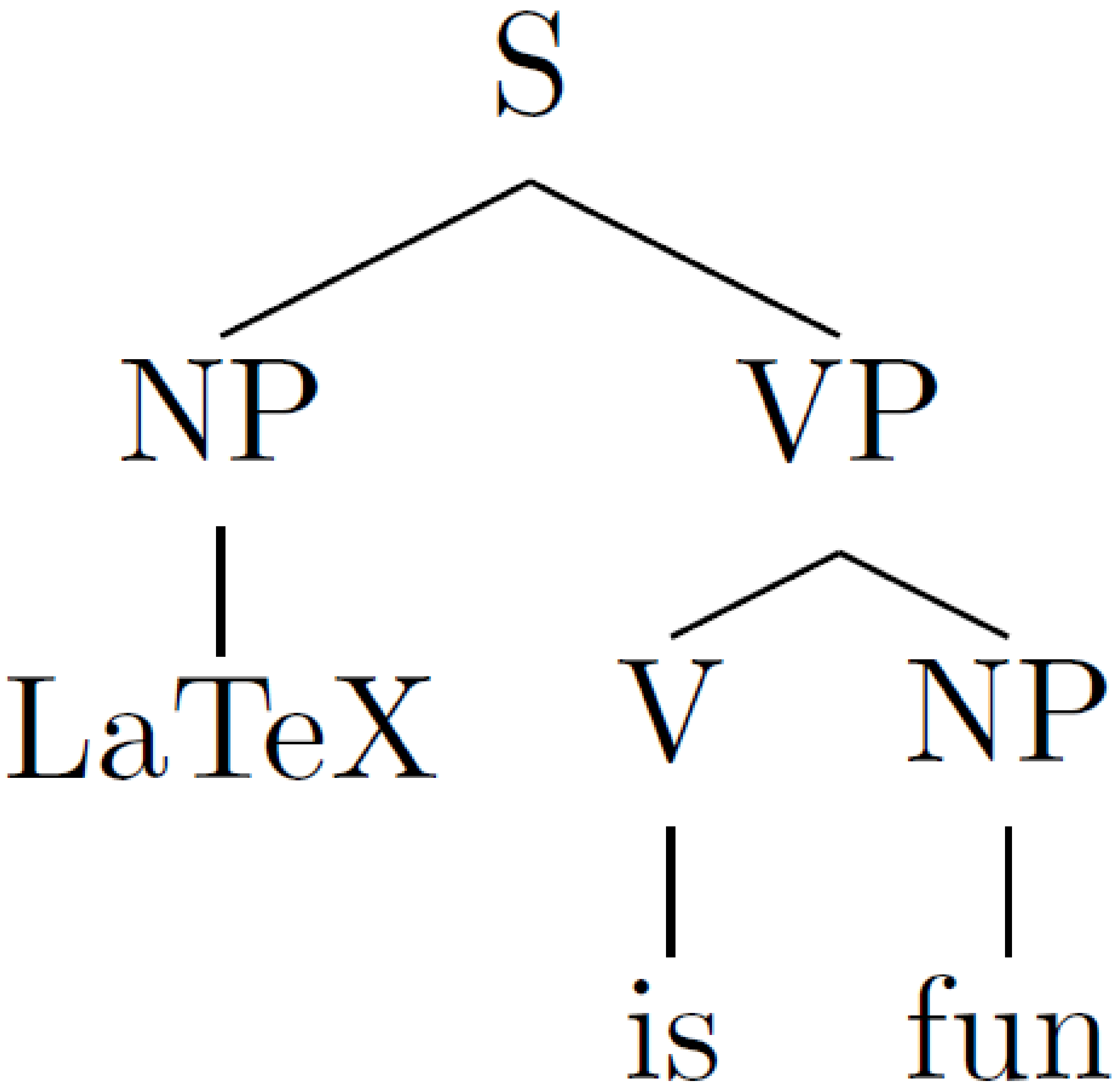


Figure 133

Note that, other than for qtree, trees are not centered by default. To center them, put them into a centered environment:

```
\begin{center}
\Tree [.S [.NP LaTeX ] [.VP [.V is ] [.NP fun ] ] ]
\end{center}
```

For setting the style of trees, tikz-qtree provides the **\tikzset** command. For example, to make a tree grow from left to right instead of from top to bottom, use the following code:

```
\tikzset{grow'=right} % make trees grow from left to right
\tikzset{every tree node/.style={anchor=base west}} % allign nodes of the tree
 to the left (west)
\Tree [.S [.NP LaTeX ] [.VP [.V is ] [.NP fun ] ] ]
```

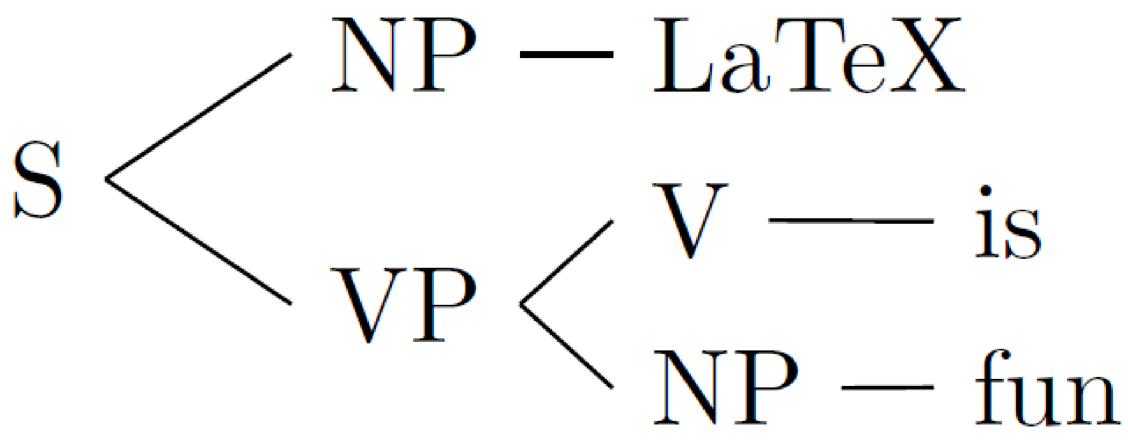


Figure 134

The above code changes the default orientation for **all** trees that are defined after **\tikzset** commands. To only change the direction of a single tree, it has to be put into a **\tikzpicture** environment:

```
\begin{tikzpicture} % all changes only affect trees within this environment
\tikzset{grow'=right} % make trees grow from left to right
\tikzset{every tree node/.style={anchor=base west}} % allign nodes of the tree
 to the left (west)
\Tree [.S [.NP LaTeX ] [.VP [.V is ] [.NP fun ] ] ]
\end{tikzpicture}
```

33.2.2. Dependency Trees

Dependency trees can take multiple visual forms. Commonly, they quite resemble phrase structure trees. Alternatively, they can be captured by brackets drawn above running text.

Two-dimensional Dependency Trees

These can be either achieved using the fairly universal drawing package TikZ, like so:

```
% In the preamble:
\usepackage{tikz}

% In the document:
\begin{tikzpicture}
        \node (is-root) {is}
                [sibling distance=3cm]
                child { node {this} }
                child {
                        node {tree}
                                [sibling distance=1.5cm]
                                child { node {an} }
                                child { node {example} }
                                child { node {.} }
                                child[missing]
                };
        \path (is-root) +(0,-2.5\tikzleveldistance)
```

```
                node {\textit{This is an example tree.}};
\end{tikzpicture}
```

which gives you the following drawing:

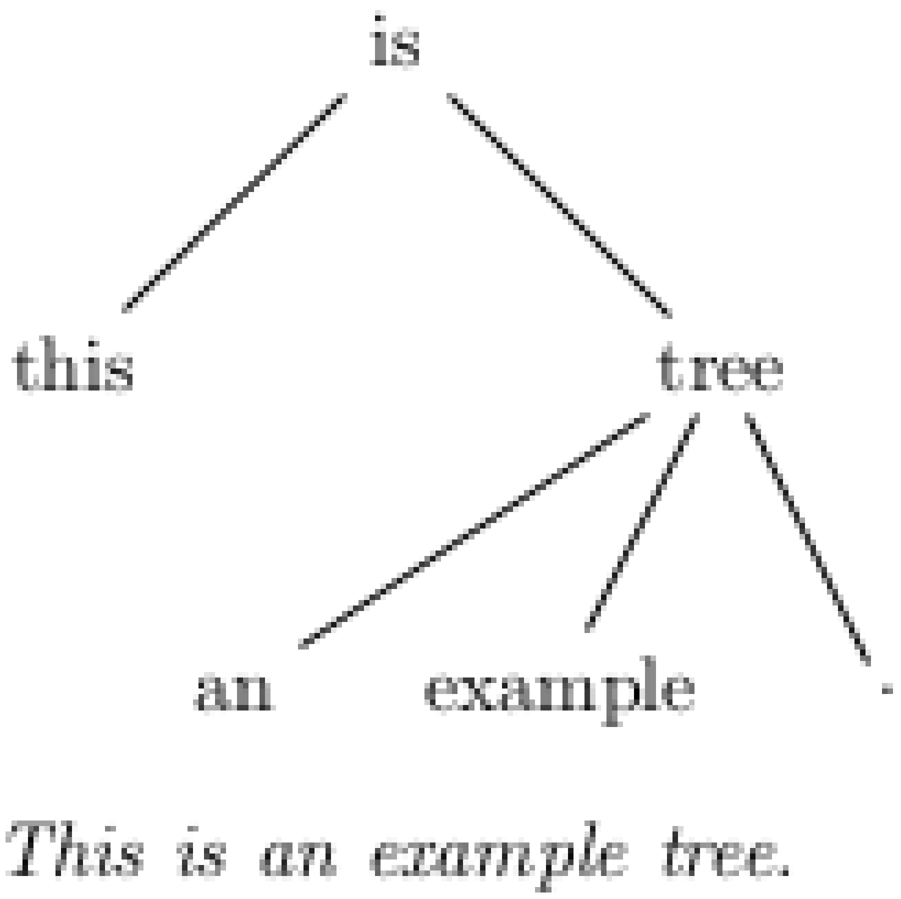


Figure 135 A dependency tree created using TikZ

TikZ has the advantage that it allows for generating PDF directly from the LaTeX source, without need for any detour of compiling to DVI using `latex` , and then converting to PDF probably via PS using tools such as `dvips` and `ps2pdf` . Latter is the case of another package based on the package **xy** , namely **xyling** .

The code for a similar tree using **xyling** might look like:

```
% In the preamble:
\usepackage{xytree}

% In the document:
\Tree{        & \K{is}\B{dl}\B{drr} \\
        \K{this} &&& \K{tree}\B{dll}\B{dl}\B{dr} \\
        & \K{an} & \K{example} && \K{.} }
```

```
\medskip
\textit{This is an example tree.}
```

which gives you a drawing like this:

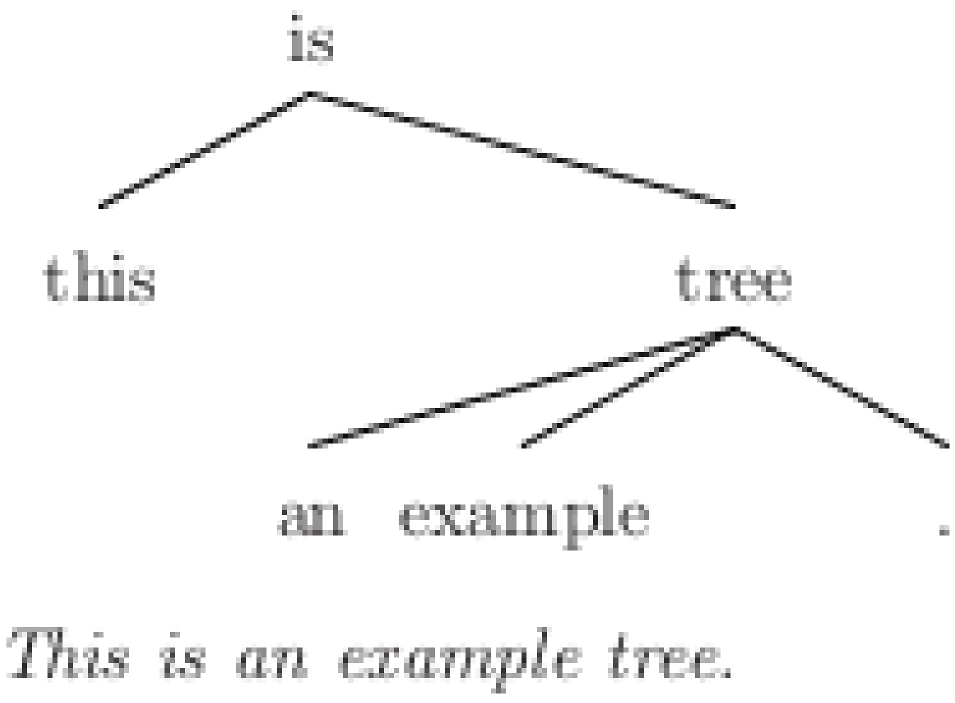


Figure 136 A dependency tree created using `xyling`

Dependency Trees as Brackets above Text

One way to typeset dependency brackets above running text is using the package **xytree** . It gives you fairly good control of how the brackets are typeset but requires compiling the LaTeX code to DVI (and perhaps converting to PDF using the tools `dvips` and `ps2pdf` later).

An example code:

```
% In the preamble:
\usepackage{xytree}

% In the document:
\xytext{
  \xybarnode{Peter} &~~~&
  \xybarnode{and}
    \xybarconnect(UL,U){-2}"_{\small conj}"
    \xybarconnect(UR,U){2}"^{\small conj}"
    &~~~&
```

```
  \xybarnode{Mary} &~~~&
  \xybarnode{bought}
    \xybarconnect[8](UL,U){-4}"_{\small subj}"
    \xybarconnect[13]{6}"^{\small punct}"
    \xybarconnect[8](UR,U){4}"^{\small obj}"
    &~~~&
  \xybarnode{a} &~~~&
  \xybarnode{car}
    \xybarconnect(UL,U){-2}"_{\small det}"
    &~~~&
  \xybarnode{.}
}
```

results in:

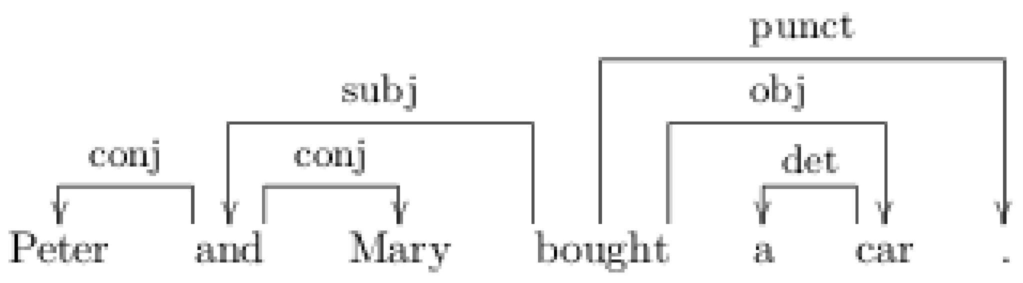


Figure 137 A dependency tree above running text created using `xytree`

Dependency Trees using TikZ-dependency

The package provides high level commands to design and style dependency graphs. To draw a graph, you only need to create a `dependency` environment, write the text of the sentence within the `deptext` environment and use `depedge` commands to draw the edges. Global and local optional parameters can be used to style and fine tune the looks of the graph, as shown in the following example:

```
% In the preamble:
\usepackage{tikz-dependency}

% In the document:
\begin{dependency}[theme = simple]
   \begin{deptext}[column sep=1em]
      A \& hearing \& is \& scheduled \& on \& the \& issue \& today \& . \\
   \end{deptext}
   \deproot{3}{ROOT}
   \depedge{2}{1}{ATT}
   \depedge[edge start x offset=-6pt]{2}{5}{ATT}
   \depedge{3}{2}{SBJ}
   \depedge{3}{9}{PU}
   \depedge{3}{4}{VC}
   \depedge{4}{8}{TMP}
   \depedge{5}{7}{PC}
   \depedge[arc angle=50]{7}{6}{ATT}
\end{dependency}
```

This code snippet would produce the following result:

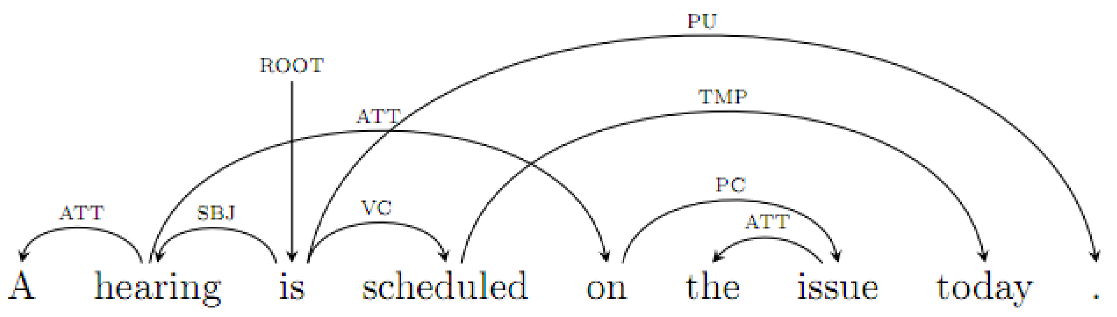


Figure 138 A dependency tree drawn with TikZ-dependency.

33.3. Glosses

Below, it is explained how to make glossed examples with different packages.

33.3.1. With `gb4e`

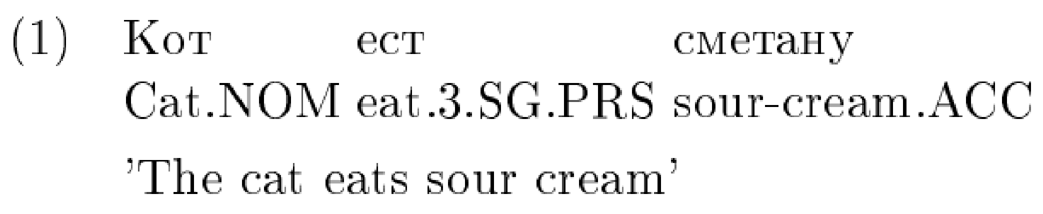
To create a glossed example, use the normal

```
exe
```

environment. But after the `\ex` tag, introduce the example and its gloss using `\gll` and the translation after it with `\trans` tag.

```
\begin{exe}
\ex
\gll Кот ест сметану\\
cat.NOM eat.3.SG.PRS sour-cream.ACC\\
\trans `The cat eats sour cream'
\end{exe}
```

The code will produce the following output:



(1) Кот ест сметану
Cat.NOM eat.3.SG.PRS sour-cream.ACC
'The cat eats sour cream'

Figure 139

Vertically aligned glosses are separated by spaces, so if it's necessary to include a space in part the gloss, simply enclose the connected parts inside braces.

```
\begin{exe}
\ex
\gll Pekka pel\"astyi karhusta.\\
     Pekka {became afraid} bear.ELA\\
\trans `Pekka became afraid because of the/a bear.'
\end{exe}
```

33.3.2. With `lingmacros`

The

```
lingmacros
```

package uses the `\shortex` command to introduce glossed examples inside the `\enumsentence` and `\eenumsentence` commands. This command takes four arguments and builds off the normal tabular[13] environment. Its first argument specifies the number of columns in the gloss. The second and third arguments give the text and its gloss respectively, and items within each column are divided by the usual `&` tabular separator. The fourth argument is the translation.

```
\enumsentence{\shortex{3}
               {Pekka & pel\"astyi & karhu-sta.}
               {Pekka & became afraid & bear.ELA}
               {`Pekka became afraid because of the/a bear.'}
               }
```

33.4. IPA characters

The

```
tipa
```

package is the standard LaTeX package for International Phonetic Alphabet symbols.

```
\usepackage{tipa}
```

There are two methods for getting IPA symbols into a document. The first way is to use the

```
IPA
```

environment.

```
\begin{IPA}
text in IPA format here
\end{IPA}
```

13 Chapter 14.6 on page 173

This method is useful for long stretches of text that need to be in IPA. Alternatively, there is the `\textipa` command that will format the text in its argument into IPA. This command is similar to other font typesetting commands[14].

```
\textipa{text in IPA format here}
```

33.4.1. Basic symbols

The IPA format works by translating ASCII characters into corresponding IPA symbols. Lower case letters are rendered as usual,

```
\textipa{abcdefghijklmnopqrstuvwxyz}
```

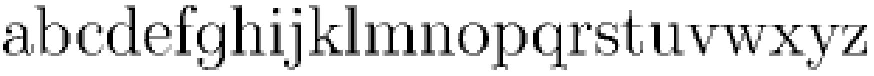

abcdefghijklmnopqrstuvwxyz

Figure 140

however capital letters are rendered differently.

```
\textipa{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
```

produces:

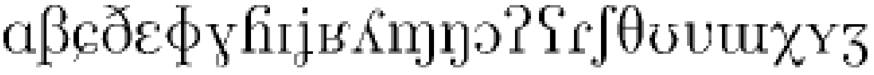

ɑβçðɛɸɣɦɪʝʁʎɱŋɔʔʕɾʃθʊʋɯχʏʒ

Figure 141

Punctuation marks that are normally used in LaTeX are also rendered faithfully in the

```
IPA
```

environment.

```
\textipa{! * + = ? . , / [ ] ( ) ` ' | ||}
```

produces:

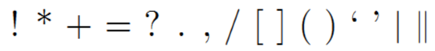

! * + = ? . , / [ ] ( ) ‘ ’ | ||

Figure 142

Numerals and `@` also have variants in the

```
tipa
```

environment.

14 `http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Fonts`

```
\textipa{1234567890 @}
```

produces:

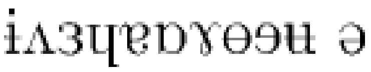


Figure 143

In addition, there are a number of special macros for representing symbols that don't have other associations, some of which are listed here. For a complete list see the official TIPA Manual[15].

The `\;` macro preceding a capital letter produces a small caps version of the letter.

```
\textipa{\;A \;B \;E \;G \;H \;I \;L \;R \;Y}
```

produces:

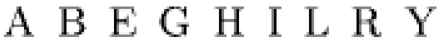


Figure 144

The `\:` macro produces retroflex symbols.

```
\textipa{\:d \:l \:n \:r \:s \:t \:z}
```

gets you:

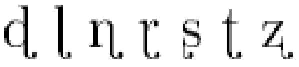


Figure 145

The `\!` macro produces implosive symbols and the bilabial click.

```
\textipa{\!b \!d \!g \!j \!G \!o}
```

gets you:

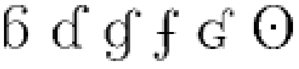


Figure 146

15 TIPA manual ^{http://mirrors.ctan.org/fonts/tipa/tipa/doc/tipaman.pdf}

33.5. References

33.6. External links

- LaTeX for Linguists[16]
- The qtree package for drawing syntactic trees.[17]
- The gb4e package page on CTAN. [18]

16 `http://www.essex.ac.uk/linguistics/external/clmt/latex4ling/`
17 `http://www.ling.upenn.edu/advice/latex/qtree/`
18 `http://www.ctan.org/tex-archive/macros/latex/contrib/gb4e/`

Part V.

Special Pages

34. Indexing

Especially useful in printed books, an index is an alphabetical list of words and expressions with the pages of the book upon which they are to be found. LaTeX supports the creation of indices with its package `makeidx`, and its support program `makeindex` , called on some systems `makeidx` .

34.1. Using `makeidx`

To enable the indexing feature of LaTeX, the `makeidx` package must be loaded in the preamble[1] with:

```
\usepackage{makeidx}
```

and the special indexing commands must be enabled by putting the

```
\makeindex
```

command into the input file preamble. This should be done within the preamble, since it tells LaTeX to create the files needed for indexing. To tell LaTeX what to index, use

```
\index{key}
```

where *key* is the index entry and does not appear in the final layout. You enter the index commands at the points in the text that you want to be referenced in the index, likely near the reason for the *key* . For example, the text

```
To solve various problems in physics, it can be advantageous
to express any arbitrary piecewise-smooth function as a
Fourier Series composed of multiples of sine and cosine functions.
```

can be re-written as

```
To solve various problems in physics, it can be advantageous
to express any arbitrary piecewise-smooth function as a Fourier Series
\index{Fourier Series}
composed of multiples of sine and cosine functions.
```

to create an entry called 'Fourier Series' with a reference to the target page. Multiple uses of *\index* with the same *key* on different pages will add those target pages to the same index entry.

To show the index within the document, merely use the command

```
\printindex
```

1 Chapter 5.2 on page 52

It is common to place it at the end of the document. The default index format is two columns.

The `showidx` package that comes with LaTeX prints out all index entries in the right margin of the text. This is quite useful for proofreading a document and verifying the index.

34.1.1. Compiling indices

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. Type in the command line:

```
makeindex   filename
```

Note that *filename* is without extension: the program will look for *filename.idx* and use that. You can optionally pass *filename.idx* directly to the program as an argument. 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 file is processed again, this sorted index gets included into the document at the point where LaTeX finds `\printindex`.

The index created by latex with the default options may not look as nice or as suitable as you would like it. To improve the looks of the index `makeindex` comes with a set of style files, usually located somewhere in the tex directory structure, usually below the `makeindex` subdirectory. To tell `makeindex` to use a specific style file, run it with the command line option:

```
makeindex   -s [style file] filename
```

If you use a GUI for compiling latex and index files, you may have to set this in the options. Here are some configuration tips for typical tools:

MakeIndex settings in WinEdt

Say you want to add an index style file named `simpleidx.ist`

- Texify/PDFTexify: Options→Execution Modes→Accessories→PDFTeXify, add to the Switches: `--mkidx-option="-s simpleidx.ist"`
- MakeIndex alone: Options→Execution Modes→Accessories→MakeIndex, add to command line: `-s simpleidx.ist`

34.1.2. Sophisticated indexing

Below are examples of `\index` entries:

Example	Index Entry	Comment
`\index{hello}`	hello, 1	Plain entry

Example	**Index Entry**	**Comment**
`\index{hello!Peter}`	Peter, 3	Subentry under 'hello'
`\index{hello!Sam@\textsl{Sam}}`	*Sam* , 2	Subentry formatted and sorted
`\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
`\index{ecole@\'ecole}`	école, 4	Handling of accents
`\index{Peter|see {hello}}`	Peter, *see* hello	Cross-references
`\index{Jen|seealso{Jenny}}`	Jen, *see also* Jenny	Same as above

Subentries

If some entry has subsections, these can be marked off with `!` . For example,

```
\index{encodings!input!cp850}
```

would create an index entry with 'cp850' categorized under 'input' (which itself is categorized into 'encodings'). These are called subsubentries and subentries in makeidx terminology.

Controlling sorting

In order to determine how an index key is sorted, place a value to sort by before the key with the `@` as a separator. This is useful if there is any formatting or math mode, so one example may be

```
\index{F@$\vec{F}$}
```

so that the entry in the index will show as '$\vec{F}$' but be sorted as 'F'.

To combine with the above feature for subentries, you should style the appropriate component(s):

```
\index{bug reports!In re code@\emph{In re} code}
```

```
\index{LaTeX@\LaTeX!Typesetting engine}
```

Changing page number style

To change the formatting of a page number, append a `|` and the name of some command which does the formatting. This command should only accept one argument.

For example, if on page 3 of a book you introduce bulldogs and include the command

```
\index{bulldog}
```

and on page 10 of the same book you wish to show the main section on bulldogs with a bold page number, use

```
\index{bulldogtextbf}
```

This will appear in the index as bulldog, 3, **10**

If you use `texindy` in place of `makeindex` , the classified entries will be sorted too, such that all the bolded entries will be placed before all others by default.

Multiple pages

To perform multi-page indexing, add a `|(` and `|)` to the end of the `\index` command, as in

```
\index{Quantum Mechanics!History(}
In 1901, Max Planck released his theory of radiation dependent on quantized
 energy.
While this explained the ultraviolet catastrophe in the spectrum of
blackbody radiation, this had far larger consequences as the beginnings of
 quantum mechanics.
...
\index{Quantum Mechanics!History)}
```

The entry in the index for the subentry 'History' will be the range of pages between the two `\index` commands.

Using special characters

In order to place values with `!` , `@` , or `|` , which are otherwise escape characters, in the index, one must quote these characters in the `\index` command by putting a double quotation mark (`"`) in front of them, and one can only place a `"` in the index by quoting it (i.e., a key for " would be `\index{""}`).

This rule does not hold for \", so to put the letter ä in the index, one may still use `\index{a@\"{a}}`.

34.2. Abbreviation list

You can make a list of abbreviations with the package `nomencl` `http://www.ctan.org/tex-archive/macros/latex/contrib/nomencl/`. You may also be interested in using the `glossaries` package described in the Glossary[2] chapter. Another option is the package `acronym` `http://www.ctan.org/pkg/acronym/`.

To enable the Nomenclature feature of LaTeX, the `nomencl` package must be loaded in the preamble with:

```
\usepackage[⟨options ⟩]{nomencl}
\makenomenclature
```

2 Chapter 35 on page 429

Issue the `\nomenclature[⟨prefix⟩]{⟨symbol⟩}{⟨description⟩}` command for each symbol you want to have included in the nomenclature list. The best place for this command is immediately after you introduce the symbol for the first time. Put `\printnomenclature` at the place you want to have your nomenclature list.

Run LaTeX 2 times then

```
makeindex  filename.nlo  -s nomencl.ist -o filename.nls
```

followed by running LaTeX once again.

To add the abbreviation list to the table of content, `intoc` option can be used when declare the `nomencl` package, i.e.

```
\usepackage[intoc]{nomencl}
```

instead of using the code in Adding Index to Table Of Contents[3] section.

The title of the list can be changed using the following command:

```
\renewcommand{\nomname}{List of Abbreviations}
```

34.3. Multiple indices

If you need multiple indices you can use the package `multind` `http://www.tex.ac.uk/cgi-bin/texfaq2html?label=multind`.

This package provides the same commands as `makeidx`, but now you also have to pass a name as the first argument to every command.

```
\usepackage{multind}
\makeindex{books}
\makeindex{authors}
...
\index{books}{A book to index}
\index{authors}{Put this author in the index}
...
\printindex{books}{The Books index}
\printindex{authors}{The Authors index}
```

34.4. Adding index to table of contents

By default, Index won't show in Table Of Contents, so you have to add it manually.

To add index as a chapter, use these commands:

```
\clearpage
\addcontentsline{toc}{chapter}{Index}
\printindex
```

If you use the book class, you may want to start it on an odd page by using `\cleardoublepage`.

3 Chapter 34.4 on page 425

34.5. International indices

If you want to sort entries that have international characters (such as ő, ą, ó, ç, etc.) you may find that the sorting ”is not quite right”. In most cases the characters are treated as special characters and end up in the same group as @, ¶ or µ. In most languages that use Latin alphabet it’s not correct.

34.5.1. Generating index

Unfortunately, current version of `xindy` and `hyperref` are incompatible. When you use `textbf` or `textit` modifiers, `texindy` will print error message:`unknown cross-reference-class `hyperindexformat'! (ignored)` and won’t add those pages to index. Workaround for this bug is described on the talk page[4].

To generate international index file you have to use `texindy` instead of `makeindex` .

xindy[5] is a much more extensible and robust indexing system than the `makeindex` system. For example, one does not need to write:

```
\index{Lin@\textbf{Lin}}
```

to get the `Lin` entry after `LAN` and before `LZA` , instead, it’s enough to write

```
\index{\textbf{Lin}}
```

But what is much more important, it can properly sort index files in many languages, not only English.

Unfortunately, generating indices ready to use by `LaTeX` using `xindy` is a bit more complicated than with `makeindex` .

First, we need to know in what encoding the `.tex` project file is saved. In most cases it will be UTF-8 or ISO-8859-1, though if you live, for example in Poland it may be ISO-8859-2 or CP-1250. Check the parameter to the `inputenc` package.

Second, we need to know which language is prominently used in our document. `xindy` can natively sort indices in Albanian, Belarusian, Bulgarian, Croatian, Czech, Danish, Dutch, English, Esperanto, Estonian, Finnish, French, Georgian, German, Greek, Gypsy, Hausa, Hebrew, Hungarian, Icelandic, Italian, Klingon, Kurdish, Latin, Latvian, Lithuanian, Macedonian, Mongolian, Norwegian, Polish, Portuguese, Romanian, Russian, Serbian Slovak, Slovenian, Sorbian, Spanish, Swedish, Turkish, Ukrainian and Vietnamese,

I don’t know if other languages have similar problems, but with Polish, if your `.tex` is saved using UTF-8, the `.ind` produced by texindy will be encoded in ISO-8859-2 if you use only `-L polish` . While it’s not a problem for entries containing polish letters, as LaTeX internally encodes all letters to plain ASCII, it is for accented letters at beginning of words, they create new index entry groups, if you have, for example an ”średnia” entry,

4 `http://en.wikibooks.org/wiki/Talk%3ALaTeX%2FIndexing%23Texindy%2C%20hyperref%20and%20textbf%2C%20textit%20modifiers`
5 `http://xindy.sourceforge.net/`

you'll get a "Ś" encoded in ISO-8859-2 `.ind` file. LaTeX doesn't like if part of the file is in UTF-8 and part is in IS-8859-2. The obvious solution (adding `-C utf8`) doesn't work, `texindy` stops with

```
ERROR: Could not find file "tex/inputenc/utf8.xdy"
```

error. The fix this, you have to load the definiton style for the headings using `-M switch` :

```
-M lang/polish/utf8
```

In the end we have to run such command:

```
texindy -L polish -M lang/polish/utf8 filename.idx
```

Additional way to fix this problem is use "iconv" to create utf8.xdy from latin2.xdy

```
iconv -f latin2 -t utf8 latin2.xdy >utf8.xdy
```

in folder

```
/usr/share/xindy/tex/inputenc
```

(You must have root privileges)

xindy in kile

To use `texindy` instead of `makeindex` in kile, you have to either redefine the MakeIndex tool in Settings → Configure Kile... → Tools → Build, or define new tool and redefine other tools to use it (for example by adding it to QuickBuild).

The `xindy` definition should look similar to this:

```
General:
 Command: texindy
 Options: -L polish -M lang/polish/utf8 -I latex '%S.idx'
Advanced:
 Type: Run Outside of Kile
 Class: Compile
 Source extension: idx
 Target extension: ind
 Target file: <empty>
 Relative dir: <empty>
 State: Editor
Menu:
 Add tool to Build menu: Compile
 Icon: the one you like
```

35. Glossary

Many technical documents use terms or acronyms unknown to the general population. It is common practice to add a glossary to make such documents more accessible.

The `glossaries` package can be used to create glossaries. It supports multiple glossaries, acronyms, and symbols. This package replaces the `glossary` package and can be used instead of the `nomencl` package.[1] Users requiring a simpler solution should consider hand-coding their entries by using the `description`[2] environment, or the `longtabu`[3] environment provided by the `tabu` package.

35.1. Jump start

Place `\usepackage{glossaries}` and `\makeglossaries` in your preamble (after `\usepackage{hyperref}` if present). Then define any number of `\newglossaryentry` and `\newacronym` glossary and acronym entries in your preamble (recommended) or before first use in your document proper. Finally add a `\printglossaries` call to locate the glossaries list within your document structure. Then pepper your writing with `\gls{mylabel}` macros (and similar) to simultaneously insert your predefined text and build the associated glossary. File processing must now include a call to `makeglossaries` followed by at least one further invocation of `latex` or `pdflatex` .

35.2. Using glossaries

To use the `glossaries` package, you have to load it explicitly:

```
\usepackage{glossaries}
```

if you wish to use `xindy`[4] (recommended) for the indexing phase, as opposed to `makeindex`[5] (the default), you need to specify the `xindy` option:

```
\usepackage[xindy]{glossaries}
```

For the glossary to show up in your Table of Contents, you need to specify the `toc` option:

```
\usepackage[toc]{glossaries}
```

1 `http://www.ctan.org/pkg/nomencl`
2 Chapter 10.4.2 on page 119
3 Chapter 14.7 on page 177
4 `http://en.wikipedia.org/wiki/xindy`
5 `http://en.wikipedia.org/wiki/MakeIndex`

See also Custom Name[6] at the bottom of this page.

Finally, place the following command in your document preamble in order to generate the glossary:

```
\makeglossaries
```

Any links in resulting glossary will not be "clickable" unless you load the `glossaries` package *after* the `hyperref` package.

In addition, users who wish to make use of `makeglossaries` will need to have Perl[7] installed — this is not normally present by default on Microsoft Windows platforms. That said, `makeglossaries` simply provides a convenient interface to `makeindex` and `xindy` and is not essential.

35.3. Defining glossary entries

To use an entry from a glossary you first need to define it. There are few ways to define an entry depending on what you define and how it is going to be used.

Note that a defined entry *won't* be included in the printed glossary *unless* it is used in the document. This enables you to create a glossary of general terms and just `\include` it in all your documents.

35.4. Defining terms

To define a term in glossary you use the `\newglossaryentry` macro:

```
\newglossaryentry{<label>}{<settings>}
```

is a unique label used to identify an entry in glossary, <settings> are comma separated `key=value` pairs used to define an entry.

For example, to define a computer entry:

```
\newglossaryentry{computer}
{
  name=computer,
  description={is a programmable machine that receives input,
               stores and manipulates data, and provides
               output in a useful format}
}
```

The above example defines an entry that has the same label and entry name. This is not always the case as the next entry will show:

```
\newglossaryentry{naiive}
{
  name=na\"{\i}ve,
  description={is a French loanword (adjective, form of naïf)
```

6 Chapter 36.0.4 on page 436
7 http://en.wikibooks.org/wiki/Perl

```
              indicating having or showing a lack of experience,
              understanding or sophistication}
}
```

When you define terms, you need to remember that they will be sorted by `makeindex` or `xindy` . While `xindy` is a bit more LaTeX aware, it does it by omitting latex macros (`\"{\i}`) thus incorrectly sorting the above example as `nave` . `makeindex` won't fare much better, because it doesn't understand TeX macros, it will interpret the word exactly as it was defined, putting it inside symbol class, before words beginning with `naa` . Therefore it's needed to extend our example and specify how to sort the word:

```
\newglossaryentry{naiive}
{
  name=na\"{\i}ve,
  description={is a French loanword (adjective, form of naïf)
              indicating having or showing a lack of experience,
              understanding or sophistication},
  sort=naive
}
```

You can also specify plural forms, if they are not formed by adding "s" (we will learn how to use them in next section):

```
\newglossaryentry{Linux}
{
  name=Linux,
  description={is a generic term referring to the family of Unix-like
              computer operating systems that use the Linux kernel},
  plural=Linuces
}
```

Or, for acronyms:

```
\newacronym[longplural={Frames per Second}]{fpsLabel}{FPS}{Frame per Second}
```

This will avoid the wrong long plural: Frame per Seconds.

So far, the glossary entries have been defined as key-value lists. Sometimes, a description is more complex than just a paragraph. For example, you may want to have multiple paragraphs, itemized lists, figures, tables, etc. For such glossary entries use the command `longnewglossaryentry` in which the description follows the key-value list. The computer entry then looks like this:

```
\longnewglossaryentry{computer}
{
  name=computer
}
  {is a programmable machine that receives input,
              stores and manipulates data, and provides
              output in a useful format}
```

35.4.1. Defining symbols

Defined entries can also be symbols:

```
\newglossaryentry{pi}
{
  name={\ensuremath{\pi}},
  description={ratio of circumference of circle to its
              diameter},
  sort=pi
}
```

You can also define both a name and a symbol:

```
\newglossaryentry{real number}
{
  name={real number},
  description={include both rational numbers, such as $42$ and
              $\frac{-23}{129}$, and irrational numbers,
              such as $\pi$ and the square root of two; or,
              a real number can be given by an infinite decimal
              representation, such as $2.4871773339\ldots$ where
              the digits continue in some way; or, the real
              numbers may be thought of as points on an infinitely
              long number line},
  symbol={\ensuremath{\mathbb{R}}}
}
```

Note that not all glossary styles show defined symbols.

35.4.2. Defining acronyms

To define a new acronym you use the `\newacronym` macro:

```
\newacronym{<label>}{<abbrv>}{<full>}
```

where is the unique label identifying the acronym, <abbrv> is the abbreviated form of the acronym and <full> is the expanded text. For example:

```
\newacronym{lvm}{LVM}{Logical Volume Manager}
```

Defined acronyms can be put in separate list if you use `acronym` package option:

```
\usepackage[acronym]{glossaries}
```

35.5. Using defined terms

When you have defined a term, you can use it in a document. There are many different commands used to refer to glossary terms.

35.5.1. General references

A general reference is used with `\gls` command. If, for example, you have glossary entries defined as those above, you might use it in this way:

```
 \Gls{naiive} people don't know about
 alternative \gls{computer} operating systems:
 \glspl{Linux}, BSDs and GNU/Hurd.
```

Naïve people don't know about alternative computer opera-

ting systems: Linuces, BSDs and GNU/Hurd.

Description of commands used in above example:

```
\gls{<label>}
```

This command prints the term associated with passed as its argument. If the `hyperref` package was loaded before `glossaries` it will also be hyperlinked to the entry in glossary.

```
\glspl{<label>}
```

This command prints the plural of the defined term, other than that it behaves in the same way as `gls`.

```
\Gls{<label>}
```

This command prints the singular form of the term with the first character converted to upper case.

```
\Glspl{<label>}
```

This command prints the plural form with first letter of the term converted to upper case.

```
\glslink{<label>}{<alternate text>}
```

This command creates the link as usual, but typesets the *alternate text* instead. It can also take several options which changes its default behavior (see the documentation).

```
\glssymbol{<label>}
```

This command prints what ever is defined in \newglossaryentry{}{symbol={Output of glssymbol}, ...}

```
\glsdesc{<label>}
```

This command prints what ever is defined in \newglossaryentry{}{description={Output of glsdesc}, ...}

35.5.2. Referring acronyms

Acronyms behave a bit differently than normal glossary terms. On first use the `\gls` command will display "<full> (<abbrv>)". On subsequent uses only the abbreviation will be displayed.

To reset the first use of an acronym use:

```
\glsreset{<label>}
```

or, if you want to reset the use status of all acronyms:

```
\glsresetall
```

If you just want to print the long version of an acronym without the abbreviation ”<full>”, use :

```
\acrlong{<label>}
```

If you just want to print the long version of an acronym with the abbreviation ”<full> (<abbrv>)”, use :

```
\acrfull{<label>}
```

If you just want to print the abbreviation ”<abbrv>”, use :

```
\acrshort{<label>}
```

36. Displaying the Glossary

To display the sorted list of terms you need to add:

```
\printglossaries
```

at the place you want the glossary and the list of acronyms to appear.

If all entries are to be printed the command

```
\glsaddall
```

can be inserted before `\printglossaries`. You may also want to use `\usepackage[nonumberlist]{glossaries}` to suppress the location list within the glossary.

36.0.3. Separate Glossary and List of Acronyms

`\printglossaries` will display all the glossaries in the order in which they were defined.[1] If no custom glossaries are defined, the default glossary and the list of acronyms will be displayed.

The glossary and the list of acronyms can be displayed separately in different places[2]:

```
\usepackage[acronym]{glossaries}

\printglossary[type=\acronymtype] % prints just the list of acronyms

Some text between the list of acronyms and the glossary.

\printglossary % if no option is supplied the default glossary is printed.
```

Dual entries with reference to a glosssary entry from an acronym

It may be useful to have both an acronym and a glossary entry for the same term. To link these two, define the acronym with a reference to the glossary entry like this:

```
\newglossaryentry{gls-OWD} {
  name={One-Way Delay},
  description={The time a packet uses through a network from one host to
 another},
}
```

1 http://mirror.ox.ac.uk/sites/ctan.org/macros/latex/contrib/glossaries/glossaries-user.html#dx1-35001

2 http://mirror.ox.ac.uk/sites/ctan.org/macros/latex/contrib/glossaries/glossaries-user.html#dx1-43001

```
\newacronym[see={[Glossary:]{gls-OWD}}]{OWD}{OWD}{One-Way
 Delay\glsadd{gls-OWD}}
```

```
Refer to acronym with \gls{OWD} and the glossary with \gls{gls-OWD}
```

To make this easier, we can use this command (modified from example in the official docs):

```
Syntax: \newdualentry[glossary options][acronym
options]{label}{abbrv}{long}{description}
```

```
\usepackage{xparse}
\DeclareDocumentCommand{\newdualentry}{ O{} O{} m m m m } {
  \newglossaryentry{gls-#3}{name={#5},text={#5\glsadd{#3}},
    description={#6},#1
  }
  \newacronym[see={[Glossary:]{gls-#3}},#2]{#3}{#4}{#5\glsadd{gls-#3}}
}
```

then, define new (dual) entries for glossary and acronym list like this:

```
\newdualentry{OWD} % label
  {OWD}            % abbreviation
  {One-Way Delay}  % long form
  {The time a packet uses through a network from one host to another} %
 description
```

36.0.4. Custom Name

The name of the glossary section can be replaced with a custom name or translated to a different language. Add the option `title` to `\printglossary` to specify the glossary's title. Add the option `toctitle` to specify a the title used in the table of content (if not used, `title` is used as default). [3]

```
\printglossary[title=List of Terms,toctitle=Terms and abbreviations]
```

36.0.5. Remove the point

To omit the dot[4] at the end of each description, use this code:

```
\usepackage[nopostdot]{glossaries}
```

36.0.6. Changing Glossary Entry Presentation Using Glossary Styles

A number of pre-built styles are available, and can be changed easily using

```
% Must be issued before \printglossaries
\glossarystyle{<newstyle>}
```

3 User Manual for glossaries.sty v4.02 as of 2014.01.13 http://mirror.ox.ac.uk/sites/ctan.org/macros/latex/contrib/glossaries/glossaries-user.html#sec:printglossary

4 http://en.wikipedia.org/wiki/Full_stop

Commonly used styles include list

My Term Has some long description 7, 9

altlist (inserts newline after term and indents description)

My Term
Has some long description 7, 9

altlistgroup or listgroup (group adds grouping based on the first letters of the terms)

M
My First Term
Has some long description 7, 9
My Second Term
Has some long description 7, 9

altlisthypergroup or listhypergroup (hyper adds an hyperlinked 'index' at the top of each glossary to jump to a group)

A|B|C|D|F|G|I|M|O|R|S|C|D|G|M|P
A
A First term
Has some long description 7, 9
B
Barely missed first
Has some long description 7, 9

36.1. Building your document

Building your document and its glossary requires three steps:

1. build your LaTeX document — this will also generate the files needed by `makeglossaries`
2. invoke `makeglossaries` — a script which selects the correct character encodings and language settings and which will also run `xindy` or `makeindex` if these are specified in your document file
3. build your LaTeX document again — to produce a document with glossary entries

Thus:

```
latex doc
makeglossaries doc
latex doc
```

where `latex` is your usual build call (perhaps `pdflatex`) and `doc` is the name of your LaTeX master file.

If your entries are interlinked (entries themselves link to other entries with `\gls` calls), you will need to run steps 1 and 2 twice, that is, in the following order: 1, 2, 1, 2, 3.

If you encounter problems, view the `doc.log` and `doc.glg` files in a text editor for clues.

37. Example for use in windows with Texmaker

37.1. Compile glossary with xindy - In Windows with Texmaker

In TeX Live `xindy` is already included, but users of MiKTeX need first do download and install `xindy` for Windows.

There are two approaches:

- The easy one.

This install program includes the Perl interpreter for `makeglossaries` , and of course `xindy` for sorting, and also adds the binaries to your windows PATH:

Download xindy, and install[1] (original source here[2]). But note: *This installs **deprecated versions** of them all!*

- The more difficult one.

See description on TeX.SE[3]: How to use Xindy with MiKTeX?[4] This way you can update yourself if necessary.

You need to restart Texmaker after installation of `xindy` , to update PATH references to `xindy` and Perl binaries.

Then, in Texmaker, go to **User** -> **User Commands** -> **Edit User Commands** .

Choose command 1

1. Menuitem = **makeglossaries**
2. Command = **makeglossaries %**

Now push **Alt+Shift+F1** and then ->**F1**

1 `http://www.fys.ku.dk/~tlinnet/xindy-win.exe`
2 `http://permalink.gmane.org/gmane.comp.text.xindy.general/743`
3 `http://tex.stackexchange.com`
4 `http://tex.stackexchange.com/q/71167/`

Note , for use with the "use build directory" option of Texmaker: makeglossaries needs to find the aux file. Thankfully, while Texmaker does not help there, the option **-d**

of makeglossaries provides for the subdirectory case. Hence the Command in this case should be:

Command = **makeglossaries -d build %** instead.

37.2. Document preamble

In preample should be included (note, **hyperref** should be loaded **before** the **glossaries**):

```
\usepackage[nomain,acronym,xindy,toc]{glossaries} % nomain, if you define
glossaries in a file, and you use \include{INP-00-glossary}
\makeglossaries
\usepackage[xindy]{imakeidx}
\makeindex
```

37.3. Glossary definitions

Write all your glossaries/acronyms in a file: Ex: **INP-00-glossary.tex** \newacronym{ddye}{D$_{\text{dye}}$}{donor dye, ex. Alexa 488} \newacronym[description={\glslink{r0}{F\"{o}rster distance}}]{R0}{R_{0}}{F\"{o}rster distance} \newglossaryentry{r0}{name=\glslink{R0}{\ensuremath{R_{0}}},text=F\"{o}rster distance,description={F\"{o}rster distance, where 50\% ...}, sort=R} \newglossaryentry{kdeac}{name=\glslink{R0}{\ensuremath{k_{DEAC}}},text=k_{DEAC}, description={is the rate of deactivation from ... and emission)}, sort=k}

37.4. Include glossary definitions and print glossary

Include glossary definitions in the preamble (Before "\begin{document}")

```
\loadglsentries[main]{INP-00-glossary}
% or using \input:
%\input{INP-00-glossary}

\begin{document}
```

Print glossaries, near end

```
\appendix
\bibliographystyle{plainnat}
\bibliography{bibtex}
\printindex
\printglossaries
\end{document}
```

37.5. References

- The `glossaries` documentation, `http://tug.ctan.org/tex-archive/macros/latex/contrib/glossaries/`
- *Using LaTeX to Write a PhD Thesis* , Nicola L.C. Talbot, `http://theoval.cmp.uea.ac.uk/~nlct/latex/thesis/node25.html`
- *glossaries FAQ* , Nicola L. C. Talbot, glossaries FAQ[5]
- *Glossaries, Nomenclature, Lists of Symbols and Acronyms* , Nicola L. C. Talbot, link[6]

5 `http://www.dickimaw-books.com/faqs/glossariesfaq.html`
6 `http://www.latex-community.org/know-how/263-glossaries-nomenclature-lists-of-symbols-and-acronyms`

38. Bibliography Management

For any academic/research writing, incorporating references into a document is an important task. Fortunately, LaTeX has a variety of features that make dealing with references much simpler, including built-in support for citing references. However, a much more powerful and flexible solution is achieved thanks to an auxiliary tool called BibTeX[1] (which comes bundled as standard with LaTeX). Recently, BibTeX has been succeeded by BibLaTeX, a tool configurable within LaTeX syntax.
BibTeX provides for the storage of all references in an external, flat-file database. (BibLaTeX uses this same syntax.) This database can be referenced in any LaTeX document, and citations made to any record that is contained within the file. This is often more convenient than embedding them at the end of every document written; a centralized bibliography source can be linked to as many documents as desired (write once, read many!). Of course, bibliographies can be split over as many files as one wishes, so there can be a file containing sources concerning topic A (`a.bib`) and another concerning topic B (`b.bib`). When writing about topic AB, both of these files can be linked into the document (perhaps in addition to sources `ab.bib` specific to topic AB).

38.1. Embedded system

If you are writing only one or two documents and aren't planning on writing more on the same subject for a long time, you might not want to waste time creating a database of references you are never going to use. In this case you should consider using the basic and simple bibliography support that is embedded within LaTeX.
LaTeX provides an environment called `thebibliography` that you have to use where you want the bibliography; that usually means at the very end of your document, just before the `\end{document}` command. Here is a practical example:

```
\begin{thebibliography}{9}

\bibitem{lamport94}
  Leslie Lamport,
  \emph{\LaTeX: a document preparation system},
  Addison Wesley, Massachusetts,
  2nd edition,
  1994.

\end{thebibliography}
```

OK, so what is going on here? The first thing to notice is the establishment of the environment. `thebibliography` is a keyword that LaTeX recognizes as everything between the begin and end tags as being data for the bibliography. The mandatory argument, which I supplied after the begin statement, is telling LaTeX how wide the item label will be when printed. Note however, that the number itself is not the parameter, but the number of digits is. Therefore, I am effectively telling LaTeX that I will only need

1 `http://www.bibtex.org`

reference labels of one character in length, which ultimately means no more than nine references in total. If you want more than nine, then input any two-digit number, such as '56' which allows up to 99 references.
Next is the actual reference entry itself. This is prefixed with the `\bibitem{`*`cite_key`*` }` command. The *cite_ key* should be a unique identifier for that particular reference, and is often some sort of mnemonic consisting of any sequence of letters, numbers and punctuation symbols (although not a comma). I often use the surname of the first author, followed by the last two digits of the year (hence *lamport94*). If that author has produced more than one reference for a given year, then I add letters after, 'a', 'b', etc. But, you should do whatever works for you. Everything after the key is the reference itself. You need to type it as you want it to be presented. I have put the different parts of the reference, such as author, title, etc., on different lines for readability. These linebreaks are ignored by LaTeX. I wanted the title to be in italics, so I used the `\emph{}` command to achieve this.

38.2. Citations

To actually cite a given document is very easy. Go to the point where you want the citation to appear, and use the following: `\cite{`*`cite_key`*` }` , where the *cite_ key* is that of the bibitem you wish to cite. When LaTeX processes the document, the citation will be cross-referenced with the bibitems and replaced with the appropriate number citation. The advantage here, once again, is that LaTeX looks after the numbering for you. If it were totally manual, then adding or removing a reference would be a real chore, as you would have to re-number all the citations by hand.

```
Instead of WYSIWYG editors, typesetting systems like \TeX{} or \LaTeX{}
 \cite{lamport94} can be used.
```

38.2.1. Referring more specifically

Sometimes you want to refer to a certain page, figure or theorem in a text book. For that you can use the arguments to the `\cite` command:

```
\cite[p.~215]{citation01}
```

The argument, "p. 215", will show up inside the same brackets. Note the tilde in [p.~215], which replaces the end-of-sentence spacing with a non-breakable inter-word space. There are two reasons: end-of-sentence spacing is too wide, and "p." should not be separated from the page number.

38.2.2. Multiple citations

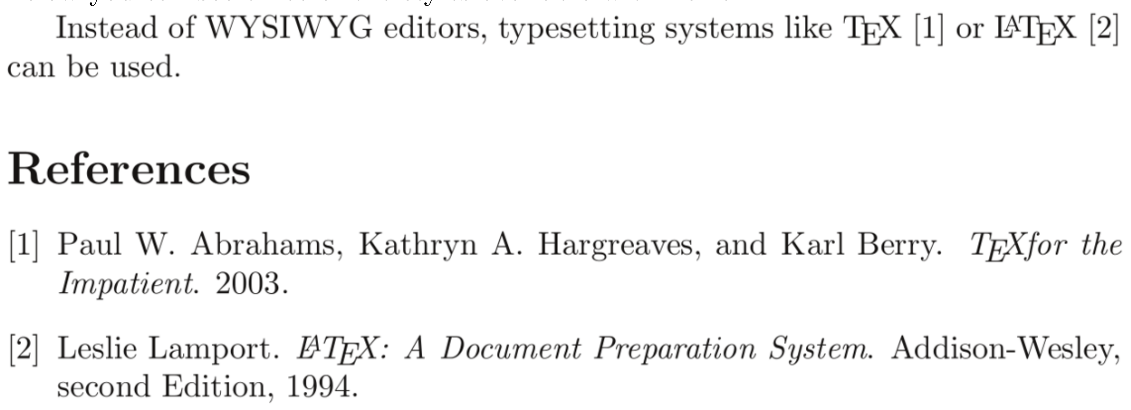
When a sequence of multiple citations are needed, you should use a single `\cite{}` command. The citations are then separated by commas. Note that you must not use spaces between the citations. Here's an example:

```
\cite{citation01,citation02,citation03}
```

The result will then be shown as citations inside the same brackets.

38.2.3. Bibliography styles

There are several different ways to format lists of bibliographic references and the citations to them in the text. These are called citation styles[2], and consist of two parts: the format of the abbreviated citation (i.e. the marker that is inserted into the text to identify the entry in the list of references) and the format of the corresponding entry in the list of references, which includes full bibliographic details.
Abbreviated citations can be of two main types: numbered or textual. Numbered citations (also known as the Vancouver referencing system[3]) are numbered consecutively in order of appearance in the text, and consist in Arabic numerals in parentheses **(1)** , square brackets **[1]** , superscript$^{\mathbf{1}}$, or a combination thereof$^{\mathbf{[1]}}$. Textual citations (also known as the Harvard referencing system[4]) use the author surname and (usually) the year as the abbreviated form of the citation, which is normally fully **(Smith**
UNKNOWN TEMPLATE time:Y
-10 years
) or partially enclosed in parenthesis, as in **Smith (**
UNKNOWN TEMPLATE time:Y
-10 years
) . The latter form allows the citation to be integrated in the sentence it supports.
Below you can see three of the styles available with LaTeX:



Instead of WYSIWYG editors, typesetting systems like TEX [1] or LATEX [2] can be used.

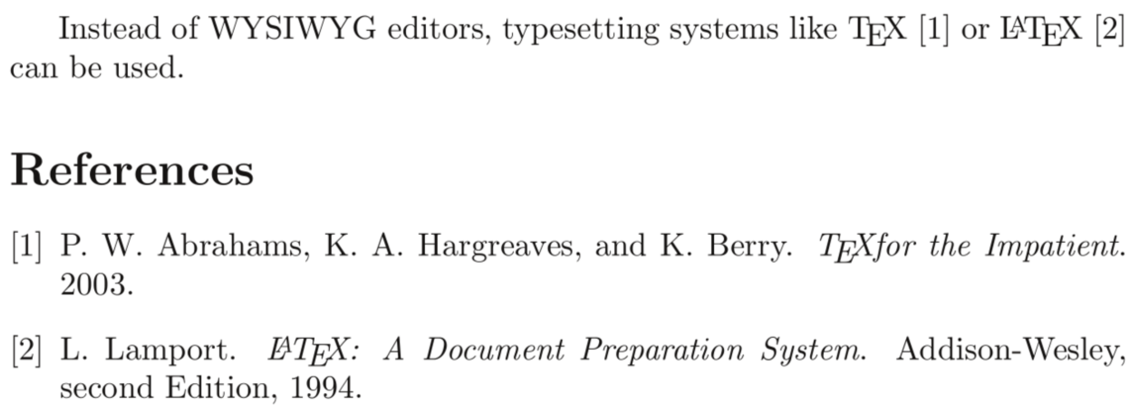
References

[1] Paul W. Abrahams, Kathryn A. Hargreaves, and Karl Berry. *TEXfor the Impatient.* 2003.

[2] Leslie Lamport. *LATEX: A Document Preparation System.* Addison-Wesley, second Edition, 1994.

Figure 147 plain

2 http://en.wikipedia.org/wiki/Citation%23Styles
3 http://en.wikipedia.org/wiki/Vancouver%20system
4 http://en.wikipedia.org/wiki/Parenthetical%20referencing



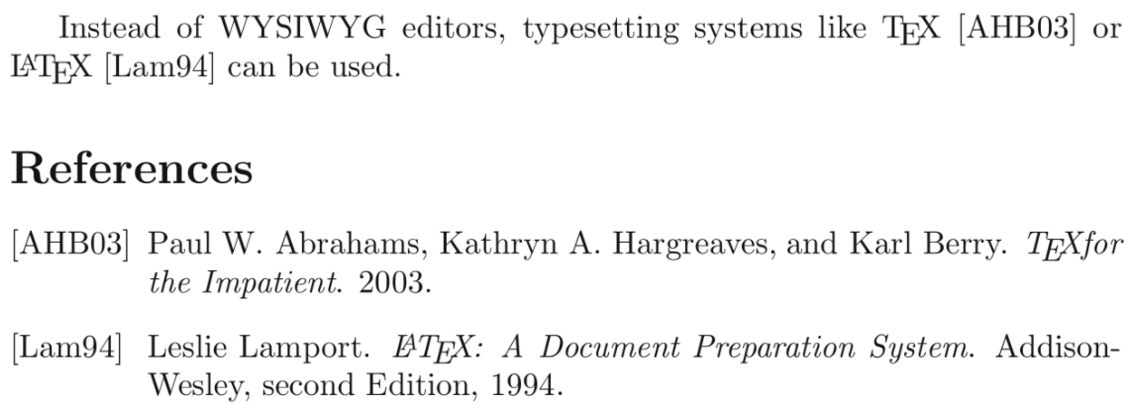
Instead of WYSIWYG editors, typesetting systems like TeX [1] or LaTeX [2] can be used.

References

[1] P. W. Abrahams, K. A. Hargreaves, and K. Berry. *TeXfor the Impatient.* 2003.

[2] L. Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, second Edition, 1994.

Figure 148 abbrv



Instead of WYSIWYG editors, typesetting systems like TeX [AHB03] or LaTeX [Lam94] can be used.

References

[AHB03] Paul W. Abrahams, Kathryn A. Hargreaves, and Karl Berry. *TeXfor the Impatient.* 2003.

[Lam94] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, second Edition, 1994.

Figure 149 alpha

Here are some more often used styles:

Style Name	**Author Name Format**	**Reference Format**	**Sorting**
plain	Homer Jay Simpson	#ID#	by author
unsrt	Homer Jay Simpson	#ID#	as referenced
abbrv	H. J. Simpson	#ID#	by author
alpha	Homer Jay Simpson	Sim95	by author
abstract	Homer Jay Simpson	Simpson-1995a	
acm	Simpson, H. J.	#ID#	
authordate1	Simpson, Homer Jay	Simpson, 1995	
apa	Simpson, H. J. (1995)	Simpson1995	
named	Homer Jay Simpson	Simpson 1995	

However, keep in mind that you will need to use the natbib package to use most of these. More examples can be found here:
- Overview of Bibtex-Styles[5]: Filterable list of styles with preview.
- Preview of several often used styles[6]

5 `http://web.archive.org/web/20131102023045id_/http://nodonn.tipido.net/bibstyle.php`
6 `http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html`

38.2.4. No cite

If you only want a reference to appear in the bibliography, but not where it is referenced in the main text, then the `\nocite{}` command can be used, for example:

```
Lamport showed in 1995 something...  \nocite{lamport95}.
```

A special version of the command, `\nocite{*}` , includes all entries from the database, whether they are referenced in the document or not.

38.2.5. Natbib

Natbib's textual and parenthetical commands	
Citation command	**Output**

`\citet{goossens93}`
`\citep{goossens93}`
	Goossens et al. (1993)
	UNKNOWN TEMPLATE red(Goossens et al.**UNKNOWN TEMPLATE red,** 1993**UNKNOWN TEMPLATE red)**

`\citet*{goossens93}`
`\citep*{goossens93}`
	Goossens, Mittlebach, and Samarin (1993)
	UNKNOWN TEMPLATE red(Goossens, Mittlebach, and Samarin, 1993**UNKNOWN TEMPLATE red)**

`\citeauthor{goossens93}`
`\citeauthor*{goossens93}`
	Goossens et al.
	GoossensUNKNOWN TEMPLATE red, Mittlebach, and Samarin

`\citeyear{goossens93}`
`\citeyearpar{goossens93}`
	1993
	UNKNOWN TEMPLATE red(1993**UNKNOWN TEMPLATE red)**

`\citealt{goossens93}`
`\citealp{goossens93}`
	Goossens et al. 1993
	Goossens et al.**UNKNOWN TEMPLATE red,** 1993

`\citetext{priv.\ comm.}`	(priv. comm.)

Using the standard LaTeX bibliography support, you will see that each reference is numbered and each citation corresponds to the numbers. The numeric style of citation is quite common in scientific writing. In other disciplines, the author-year style, e.g., (Roberts, 2003), such as *Harvard* is preferred. A discussion about which is best will not occur here, but a possible way to get such an output is by the `natbib` package. In fact, it can supersede LaTeX's own citation commands, as Natbib allows the user to easily switch between Harvard or numeric.

The first job is to add the following to your preamble in order to get LaTeX to use the Natbib package:

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

Also, you need to change the bibliography style file to be used, so edit the appropriate line at the bottom of the file so that it reads: `\bibliographystyle{plainnat}` . Once done, it is basically a matter of altering the existing `\cite` commands to display the type of citation you want.

Citation styles compatible with Natbib		
Style	**Source**	**Description**
plainnat	Provided	natbib-compatible version of plain
abbrvnat	Provided	natbib-compatible version of abbrv
unsrtnat	Provided	natbib-compatible version of unsrt
apsrev	REVTeX 4 home page[7]	natbib-compatible style for Physical Review journals
rmpaps	REVTeX 4 home page[8]	natbib-compatible style for Review of Modern Physics journals
IEEEtranN	TeX Catalogue entry[9]	natbib-compatible style for IEEE publications
achemso	TeX Catalogue entry[10]	natbib-compatible style for American Chemical Society journals
rsc	TeX Catalogue entry[11]	natbib-compatible style for Royal Society of Chemistry journals

Customization

Natbib's customization options	
Option	**Meaning**
`round` : `square` : `curly` : `angle`	Parentheses () (default), square brackets [], curly braces {} or angle brackets <>
`colon` : `comma`	multiple citations are separated by semi-colons (default) or commas
`authoryear` : `numbers` : `super`	author year style citations (default), numeric citations or superscripted numeric citations
`sort` : `sort&compress`	multiple citations are sorted into the order in which they appear in the references section or also compressing multiple numeric citations where possible
`longnamesfirst`	the first citation of any reference will use the starred variant (full author list), subsequent citations will use the abbreviated *et al.* style
`sectionbib`	for use with the chapterbib package. redefines \thebibliography to issue \section* instead of \chapter*

7 `http://authors.aps.org/revtex4/`
8 `http://authors.aps.org/revtex4/`
9 `http://www.ctan.org/tex-archive/help/Catalogue/entries/ieeetran.html`
10 `http://www.ctan.org/tex-archive/help/Catalogue/entries/achemso.html`
11 `http://www.ctan.org/tex-archive/help/Catalogue/entries/rsc.html`

Natbib's customization options	
Option	**Meaning**
`nonamebreak`	keeps all the authors' names in a citation on one line to fix some hyperref problems - causes overfull hboxes

The main commands simply add a *t* for 'textual' or *p* for 'parenthesized', to the basic `\cite` command. You will also notice how Natbib by default will compress references with three or more authors to the more concise *1st surname et al* version. By adding an asterisk (*), you can override this default and list all authors associated with that citation. There are some other specialized commands that Natbib supports, listed in the table here. Keep in mind that for instance `abbrvnat` does not support `\citet*` and will automatically choose between all authors and et al..
The final area that I wish to cover about Natbib is customizing its citation style. There is a command called `\bibpunct` that can be used to override the defaults and change certain settings. For example, I have put the following in the preamble:

```
\bibpunct{(}{)}{;}{a}{,}{,}
```

The command requires six mandatory parameters.

1. The symbol for the opening bracket.
2. The symbol for the closing bracket.
3. The symbol that appears between multiple citations.
4. This argument takes a letter:
   - *n* - numerical style.
   - *s* - numerical superscript style.
   - *any other letter* - author-year style.
5. The punctuation to appear between the author and the year (in parenthetical case only).
6. The punctuation used between years, in multiple citations when there is a common author. e.g., (Chomsky 1956, 1957). If you want an extra space, then you need `{,~}` .

Some of the options controlled by `\bibpunct` are also accessible by passing options to the natbib package when it is loaded. These options also allow some other aspect of the bibliography to be controlled, and can be seen in the table (right).
So as you can see, this package is quite flexible, especially as you can easily switch between different citation styles by changing a single parameter. Do have a look at the Natbib manual[12], it's a short document and you can learn even more about how to use it.

38.3. BibTeX

I have previously introduced the idea of embedding references at the end of the document, and then using the `\cite` command to cite them within the text. In this tutorial, I want to do a little better than this method, as it's not as flexible as it could be. I will concentrate on using BibTeX[13].

12 `http://www.ctex.org/documents/packages/bibref/natbib.pdf`
13 `http://en.wikipedia.org/wiki/BibTeX`

A BibTeX database is stored as a *.bib* file. It is a plain text file, and so can be viewed and edited easily. The structure of the file is also quite simple. An example of a BibTeX entry:

```
@article{greenwade93,
    author  = "George D. Greenwade",
    title   = "The {C}omprehensive {T}ex {A}rchive {N}etwork ({CTAN})",
    year    = "1993",
    journal = "TUGBoat",
    volume  = "14",
    number  = "3",
    pages   = "342--351"
}
```

Each entry begins with the declaration of the reference type, in the form of `@`*`type`* . BibTeX knows of practically all types you can think of, common ones are: *book* , *article* , and for papers presented at conferences, there is *inproceedings* . In this example, I have referred to an article within a journal.
After the type, you must have a left curly brace '`{` ' to signify the beginning of the reference attributes. The first one follows immediately after the brace, which is the *citation key* , or the *BibTeX key* . This key must be unique for all entries in your bibliography. It is this identifier that you will use within your document to cross-reference it to this entry. It is up to you as to how you wish to label each reference, but there is a loose standard in which you use the author's surname, followed by the year of publication. This is the scheme that I use in this tutorial.
Next, it should be clear that what follows are the relevant fields and data for that particular reference. The field names on the left are BibTeX keywords[14]. They are followed by an equals sign (=) where the value for that field is then placed. BibTeX expects you to explicitly label the beginning and end of each value. I personally use quotation marks ("), however, you also have the option of using curly braces ('{', '}'). But as you will soon see, curly braces have other roles, within attributes, so I prefer not to use them for this job as they can get more confusing. A notable exception is when you want to use characters with umlauts (ü, ö, etc), since their notation[15] is in the format `\"{o}` , and the quotation mark will close the one opening the field, causing an error in the parsing of the reference. Using `\usepackage[utf8]{inputenc}` in the preamble to the `.tex` source file can get round this, as the accented characters can just be stored in the `.bib` file without any need for special markup. This allows a consistent format to be kept throughout the `.bib` file, avoiding the need to use braces when there are umlauts to consider.
Remember that each attribute must be followed by a comma to delimit one from another. You do not need to add a comma to the last attribute, since the closing brace will tell BibTeX that there are no more attributes for this entry, although you won't get an error if you do.
It can take a while to learn what the reference types are, and what fields each type has available (and which ones are required or optional, etc). So, look at this entry type reference[16] and also this field reference[17] for descriptions of all the fields. It may be worth bookmarking or printing these pages so that they are easily at hand when you

14 `http://en.wikipedia.org/wiki/BibTeX%23Bibliographic%20information%20file`
15 `http://en.wikibooks.org/wiki/..%2FAccents`
16 `http://newton.ex.ac.uk/tex/pack/bibtex/btxdoc/node6.html`
17 `http://newton.ex.ac.uk/tex/pack/bibtex/btxdoc/node7.html`

need them. Much of the information contained therein is repeated in the following table for your convenience.

Standard BibTeX entry and field types

	article	**book**	**booklet**	**inbook**	**incollection**	**inproceedings** ≈conference	**manual**	**mastersthesis, phdthesis**	**misc**	**proceedings**	**tech report**	**unpublished**
address		o	o	o	o	o	o	o		o	o	
annote												
author	+	*	o	$*^1$	+	+	o	+	o		+	+
booktitle					+	+						
chapter				$*^2$	o							
crossref												
edition		o		o	o		o					
editor		*		$*^1$	o	o				o		
howpublished			o						o			
institution											+	
journal	+											
key												
month	o	o	o	o	o	o	o	o	o	o	o	o
note	o	o	o	o	o	o	o	o	o	o	o	+
number	o	o		o	o	o				o	o	
organization						o	o			o		
pages	o			$*^2$	o	o						
publisher		+		+	+	o				o		
school								+				
series		o		o	o	o				o		
title	+	+	+	+	+	+	+	+	o	+	+	+
type				o	o			o			o	
volume	o	o		o	o	o				o		
year	+	+	o	+	+	+	o	+	o	+	+	o

+ Required fields, O Optional fields

38.3.1. Authors

BibTeX can be quite clever with names of authors. It can accept names in *forename surname* or *surname, forename* . I personally use the former, but remember that the order you input them (or any data within an entry for that matter) is customizable and so you can get BibTeX to manipulate the input and then output it however you like. If you use the *forename surname* method, then you must be careful with a few special names, where there are compound surnames, for example ”John von Neumann”. In this form, BibTeX assumes that the last word is the surname, and everything before is the forename, plus any middle names. You must therefore manually tell BibTeX to keep the ’von’ and ’Neumann’ together. This is achieved easily using curly braces. So the final result would be ”John {von Neumann}”. This is easily avoided with the *surname, forename* , since you have a comma to separate the surname from the forename.
Secondly, there is the issue of how to tell BibTeX when a reference has more than one author. This is very simply done by putting the keyword *and* in between every author. As we can see from another example:

```
@book{goossens93,
    author    = "Michel Goossens and Frank Mittelbach and Alexander Samarin",
    title     = "The LaTeX Companion",
    year      = "1993",
    publisher = "Addison-Wesley",
    address   = "Reading, Massachusetts"
}
```

This book has three authors, and each is separated as described. Of course, when BibTeX processes and outputs this, there will only be an ’and’ between the penultimate and last authors, but within the .bib file, it needs the *and* s so that it can keep track of the individual authors.

38.3.2. Standard templates

Be careful if you copy the following templates, the % sign is not valid to comment out lines in bibtex files. If you want to comment out a line, you have to put it outside the entry.

@article

An article from a magazine or a journal.

- Required fields: author, title, journal, year.
- Optional fields: volume, number, pages, month, note.

```
@article{Xarticle,
    author    = "",
    title     = "",
    journal   = "",
    %volume   = "",
    %number   = "",
    %pages    = "",
    year      = "XXXX",
    %month    = "",
    %note     = "",
}
```

@book

A published book

- Required fields: author/editor, title, publisher, year.
- Optional fields: volume/number, series, address, edition, month, note.

```
@book{Xbook,
    author    = "",
    title     = "",
    publisher = "",
    %volume   = "",
    %number   = "",
    %series   = "",
    %address  = "",
    %edition  = "",
    year      = "XXXX",
    %month    = "",
    %note     = "",
}
```

@booklet

A bound work without a named publisher or sponsor.

- Required fields: title.
- Optional fields: author, howpublished, address, month, year, note.

```
@booklet{Xbooklet,
    %author   = "",
    title     = "",
    %howpublished   = "",
    %address  = "",
    year      = "XXXX",
    %month    = "",
    %note     = "",
}
```

@conference

Equal to inproceedings

- Required fields: author, title, booktitle, year.
- Optional fields: editor, volume/number, series, pages, address, month, organization, publisher, note.

```
@conference{Xconference,
    author    = "",
    title     = "",
    booktitle = "",
    %editor   = "",
    %volume   = "",
    %number   = "",
    %series   = "",
    %pages    = "",
    %address  = "",
    year      = "XXXX",
    %month    = "",
    %publisher= "",
    %note     = "",
}
```

@inbook

A section of a book *without* its own title.

- Required fields: author/editor, title, chapter and/or pages, publisher, year.
- Optional fields: volume/number, series, type, address, edition, month, note.

```
@inbook{Xinbook,
        author        = "",
        editor        = "",
        title        = "",
        chapter        = "",
        pages        = "",
        publisher= "",
        %volume       = "",
        %number       = "",
        %series       = "",
        %type       = "",
        %address= "",
        %edition= "",
        year       = "",
        %month       = "",
        %note       = "",
}
```

@incollection

A section of a book having its own title.

- Required fields: author, title, booktitle, publisher, year.
- Optional fields: editor, volume/number, series, type, chapter, pages, address, edition, month, note.

```
@incollection{Xincollection,
        author       = "",
        title       = "",
        booktitle= "",
        publisher= "",
        %editor       = "",
        %volume       = "",
        %number       = "",
        %series       = "",
        %type       = "",
        %chapter= "",
        %pages       = "",
        %address= "",
        %edition= "",
        year       = "",
        %month       = "",
        %note       = "",
}
```

@inproceedings

An article in a conference proceedings.

- Required fields: author, title, booktitle, year.
- Optional fields: editor, volume/number, series, pages, address, month, organization, publisher, note.

```
@inproceedings{Xinproceedings,
        author              = "",
        title              = "",
        booktitle       = "",
        %editor              = "",
        %volume              = "",
        %number              = "",
        %series              = "",
        %pages              = "",
        %address       = "",
        %organization       = "",
        %publisher       = "",
        year              = "",
        %month              = "",
        %note              = "",
}
```

@manual

Technical manual

- Required fields: title.
- Optional fields: author, organization, address, edition, month, year, note.

```
@manual{Xmanual,
        title               = "",
        %author               = "",
        %organization       = "",
        %address       = "",
        %edition       = "",
        year               = "",
        %month               = "",
        %note               = "",
}
```

@mastersthesis

Master's thesis

- Required fields: author, title, school, year.
- Optional fields: type (eg. "diploma thesis"), address, month, note.

```
@mastersthesis{Xthesis,
    author    = "",
    title     = "",
    school    = "",
    %type     = "diploma thesis",
    %address  = "",
    year      = "XXXX",
    %month    = "",
    %note     = "",
}
```

@misc

Template useful for other kinds of publication

- Required fields: none
- Optional fields: author, title, howpublished, month, year, note.

```
@misc{Xmisc,
    %author     = "",
    %title      = "",
    %howpublished = "",
    %year      = "XXXX",
    %month     = "",
    %note      = "",
}
```

@phdthesis

Ph.D. thesis

- Required fields: author, title, year, school.
- Optional fields: address, month, keywords, note.

```
@phdthesis{Xphdthesis,
        author               = "",
        title               = "",
        school               = "",
        %address       = "",
        year               = "",
        %month               = "",
        %keywords       = "",
        %note               = "",
}
```

@proceedings

The proceedings of a conference.

- Required fields: title, year.
- Optional fields: editor, volume/number, series, address, month, organization, publisher, note.

```
@proceedings{Xproceedings,
        title               = "",
        %editor               = "",
        %volume               = "",
        %number               = "",
        %series               = "",
        %address        = "",
        %organization       = "",
        %publisher        = "",
        year                = "",
        %month                = "",
        %note               = "",
}
```

@techreport
Technical report from educational, commercial or standardization institution.

- Required fields: author, title, institution, year.
- Optional fields: type, number, address, month, note.

```
@techreport{Xtreport,
    author     = "",
    title      = "",
    institution = "",
    %type      = "",
    %number    = "",
    %address   = "",
    year       = "XXXX",
    %month     = "",
    %note      = "",
}
```

@unpublished
An unpublished article, book, thesis, etc.

- Required fields: author, title, note.
- Optional fields: month, year.

```
@unpublished{Xunpublished,
        author        = "",
        title        = "",
        %year        = "",
        %month        = "",
        note        = "",
}
```

38.3.3. Not standard templates

@patent
BiBTeX entries can be exported from Google Patents.
(see Cite Patents with Bibtex[18] for an alternative)
@collection
@electronic

38.3.4. Preserving case of letters

In the event that BibTeX has been set by the chosen style to not preserve all capitalization within titles, problems can occur, especially if you are referring to proper nouns, or acronyms. To tell BibTeX to keep them, use the good old curly braces around the letter

18 http://www.see-out.com/sandramau/bibpat.html

in question, (or letters, if it's an acronym) and all will be well! It is even possible that lower-case letters may need to be preserved - for example if a chemical formula is used in a style that sets a title in all caps or small caps, or if "pH" is to be used is a style that capitalises all first letters.

```
title = "The {LaTeX} Companion",
```

However, **avoid** putting the whole title in curly braces, as it will look odd if a different capitalization format is used:

```
title = "{The LaTeX Companion}",
```

For convenience though, many people simply put double curly braces, which may help when writing scientific articles for different magazines, conferences with different BibTex styles that do sometimes keep and sometimes not keep the capital letters:

```
title = {{The LaTeX Companion}},
```

As an alternative, try other BibTex styles or modify the existing. The approach of putting only relevant text in curly brackets is the most feasible if using a template under the control of a publisher, such as for journal submissions. Using curly braces around single letters is also to be avoided if possible, as it may mess up the kerning, especially with biblatex,[19] so the first step should generally be to enclose single words in braces.

38.3.5. A few additional examples

Below you will find a few additional examples of bibliography entries. The first one covers the case of multiple authors in the Surname, Firstname format, and the second one deals with the incollection case.

```
@article{AbedonHymanThomas2003,
  author = "Abedon, S. T. and Hyman, P. and Thomas, C.",
  year = "2003",
  title = "Experimental examination of bacteriophage latent-period evolution as
 a response to bacterial availability",
  journal = "Applied and Environmental Microbiology",
  volume = "69",
  pages = "7499--7506"
},

@incollection{Abedon1994,
  author = "Abedon, S. T.",
  title = "Lysis and the interaction between free phages and infected cells",
  pages = "397--405",
  booktitle = "Molecular biology of bacteriophage T4",
  editor = "Karam, Jim D. Karam and Drake, John W. and Kreuzer, Kenneth N. and
 Mosig, Gisela
           and Hall, Dwight and Eiserling, Frederick A. and Black, Lindsay W.
 and Kutter, Elizabeth
           and Carlson, Karin and Miller, Eric S. and Spicer, Eleanor",
  publisher = "ASM Press, Washington DC",
  year = "1994"
},
```

If you have to cite a website you can use @misc, for example:

```
@misc{website:fermentas-lambda,
      author = "Fermentas Inc.",
      title = "Phage Lambda: description \& restriction map",
```

19 The biblatex manual ˆ{http://mirrors.ctan.org/macros/latex/contrib/biblatex/doc/biblatex.pdf}

```
      month = "November",
      year = "2008",
      url = "http://www.fermentas.com/techinfo/nucleicacids/maplambda.htm"
},
```

The note field comes in handy if you need to add unstructured information, for example that the corresponding issue of the journal has yet to appear:

```
@article{blackholes,
      author="Rabbert Klein",
      title="Black Holes and Their Relation to Hiding Eggs",
      journal="Theoretical Easter Physics",
      publisher="Eggs Ltd.",
      year="2010",
      note="(to appear)"
}
```

38.3.6. Getting current LaTeX document to use your .bib file

At the end of your LaTeX file (that is, after the content, but before `\end{document}`), you need to place the following commands:

```
\bibliographystyle{plain}
\bibliography{sample1,sample2,...,samplen}
% Note the lack of whitespace between the commas and the next bib file.
```

Bibliography styles are files recognized by BibTeX that tell it how to format the information stored in the `.bib` file when processed for output. And so the first command listed above is declaring which style file to use. The style file in this instance is `plain.bst` (which comes as standard with BibTeX). You do not need to add the .bst extension when using this command, as it is assumed. Despite its name, the plain style does a pretty good job (look at the output of this tutorial to see what I mean).
The second command is the one that actually specifies the `.bib` file you wish to use. The ones I created for this tutorial were called `sample1.bib` , `sample2.bib` , . . ., `samplen.bib` , but once again, you don't include the file extension. At the moment, the `.bib` file is in the same directory as the LaTeX document too. However, if your .bib file was elsewhere (which makes sense if you intend to maintain a centralized database of references for all your research), you need to specify the path as well, e.g `\bibliography{/some/where/sample}` or `\bibliography{../sample1}` (if the `.bib` file is in the parent directory of the `.tex` document that calls it).
Now that LaTeX and BibTeX know where to look for the appropriate files, actually citing the references is fairly trivial. The `\cite{`*`ref_key`*` }` is the command you need, making sure that the *ref_ key* corresponds exactly to one of the entries in the .bib file. If you wish to cite more than one reference at the same time, do the following: `\cite{`*`ref_key1 , ref_key2 , ..., ref_keyN`*` }` .

38.3.7. Why won't LaTeX generate any output?

The addition of BibTeX adds extra complexity for the processing of the source to the desired output. This is largely hidden from the user, but because of all the complexity of the referencing of citations from your source LaTeX file to the database entries in another file, you actually need multiple passes to accomplish the task. This means you have to run LaTeX a number of times. Each pass will perform a particular task until it has managed to resolve all the citation references. Here's what you need to type (into command line):

1. `latex latex_source_code.tex`

2. `bibtex latex_source_code.aux`
3. `latex latex_source_code.tex`
4. `latex latex_source_code.tex`

(Extensions are optional, if you put them note that the bibtex command takes the AUX file as input.)
After the first LaTeX run, you will see errors such as:

```
LaTeX Warning: Citation `lamport94' on page 1 undefined on input line 21.
...
LaTeX Warning: There were undefined references.
```

The next step is to run bibtex on that same LaTeX source (or more precisely the corresponding AUX file, however not on the actual .bib file) to then define all the references within that document. You should see output like the following:

```
This is BibTeX, Version 0.99c (Web2C 7.3.1)
The top-level auxiliary file: latex_source_code.aux
The style file: plain.bst
Database file #1: sample.bib
```

The third step, which is invoking LaTeX for the second time will see more errors like "`LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.` ". Don't be alarmed, it's almost complete. As you can guess, all you have to do is follow its instructions, and run LaTeX for the third time, and the document will be output as expected, without further problems.
If you want a pdf output instead of a dvi output you can use `pdflatex` instead of `latex` as follows:

1. `pdflatex latex_source_code.tex`
2. `bibtex latex_source_code.aux`
3. `pdflatex latex_source_code.tex`
4. `pdflatex latex_source_code.tex`

(Extensions are optional, if you put them note that the bibtex command takes the AUX file as input.)
Note that if you are editing your source in vim and attempt to use command mode and the current file shortcut (%) to process the document like this:

1. `:! pdflatex %`
2. `:! bibtex %`

You will get an error similar to this:

1. `I couldn't open file name 'current_file.tex.aux'`

It appears that the file extension is included by default when the current file command (%) is executed. To process your document from within vim, you must explicitly name the file without the file extension for bibtex to work, as is shown below:

1. `:! pdflatex %`
2. `:! bibtex %:r` (without file extension, it looks for the AUX file as mentioned above)
3. `:! pdflatex %`
4. `:! pdflatex %`

However, it is much easier to install the Vim-LaTeX plugin from here[20]. This allows you to simply type \ll when not in insert mode, and all the appropriate commands are automatically executed to compile the document. Vim-LaTeX even detects how many

20 `http://vim-latex.sourceforge.net/`

times it has to run pdflatex, and whether or not it has to run bibtex. This is just one of the many nice features of Vim-LaTeX, you can read the excellent Beginner's Tutorial[21] for more about the many clever shortcuts Vim-LaTeX provides.
Another option exists if you are running Unix/Linux or any other platform where you have make[22]. Then you can simply create a Makefile and use vim's make command or use make in shell. The Makefile would then look like this:

```
latex_source_code.pdf: latex_source_code.tex latex_source_code.bib
    pdflatex latex_source_code.tex
    bibtex latex_source_code.aux
    pdflatex latex_source_code.tex
    pdflatex latex_source_code.tex
```

38.3.8. Including URLs in bibliography

As you can see, there is no field for URLs. One possibility is to include Internet addresses in `howpublished` field of `@misc` or `note` field of `@techreport` , `@article` , `@book` :

```
howpublished = "\url{http://www.example.com}"
```

Note the usage of `\url` command to ensure proper appearance of URLs[23].
Another way is to use special field `url` and make bibliography style recognise it.

```
url = "http://www.example.com"
```

You need to use `\usepackage{url}` in the first case or `\usepackage{hyperref}` in the second case.
Styles provided by Natbib (see below) handle this field, other styles can be modified using urlbst[24] program. Modifications of three standard styles (plain, abbrv and alpha) are provided with urlbst.
If you need more help about URLs in bibliography, visit FAQ of UK List of TeX[25].

38.3.9. Customizing bibliography appearance

One of the main advantages of BibTeX, especially for people who write many research papers, is the ability to customize your bibliography to suit the requirements of a given publication. You will notice how different publications tend to have their own style of formatting references, to which authors must adhere if they want their manuscripts published. In fact, established journals and conference organizers often will have created their own bibliography style (.bst file) for those users of BibTeX, to do all the hard work for you.
It can achieve this because of the nature of the .bib database, where all the information about your references is stored in a structured format, but nothing about style. This is a common theme in LaTeX in general, where it tries as much as possible to keep content and presentation separate.

21 http://vim-latex.sourceforge.net/documentation/latex-suite-quickstart/
22 http://en.wikipedia.org/wiki/Make_%28software%29
23 http://en.wikibooks.org/wiki/LaTeX%2FFormatting%23Typesetting%20URLs
24 http://purl.org/nxg/dist/urlbst
25 http://www.tex.ac.uk/cgi-bin/texfaq2html?label=citeURL

A bibliography style file (`.bst`) will tell LaTeX how to format each attribute, what order to put them in, what punctuation to use in between particular attributes etc. Unfortunately, creating such a style by hand is not a trivial task. Which is why `Makebst` (also known as *custom-bib*) is the tool we need.
`Makebst` can be used to automatically generate a .bst file based on your needs. It is very simple, and actually asks you a series of questions about your preferences. Once complete, it will then output the appropriate style file for you to use.
It should be installed with the LaTeX distribution (otherwise, you can download it[26]) and it's very simple to initiate. At the command line, type:

```
latex makebst
```

LaTeX will find the relevant file and the questioning process will begin. You will have to answer quite a few (although, note that the default answers are pretty sensible), which means it would be impractical to go through an example in this tutorial. However, it is fairly straight-forward. And if you require further guidance, then there is a comprehensive manual[27] available. I'd recommend experimenting with it and seeing what the results are when applied to a LaTeX document.
If you are using a custom built .bst file, it is important that LaTeX can find it! So, make sure it's in the same directory as the LaTeX source file, *unless* you are using one of the standard style files (such as *plain* or *plainnat* , that come bundled with LaTeX - these will be automatically found in the directories that they are installed. Also, make sure the name of the `.bst` file you want to use is reflected in the `\bibliographystyle{style}` command (but don't include the `.bst` extension!).

38.3.10. Localizing bibliography appearance

When writing documents in languages other than English, you may find it desirable to adapt the appearance of your bibliography to the document language. This concerns words such as *editors* , *and* , or *in* as well as a proper typographic layout. The `babelbib` package[28] can be used here. For example, to layout the bibliography in German, add the following to the header:

```
\usepackage[fixlanguage]{babelbib}
\selectbiblanguage{german}
```

Alternatively, you can layout each bibliography entry according to the language of the cited document:

```
\usepackage{babelbib}
```

The language of an entry is specified as an additional field in the BibTeX entry:

```
@article{mueller08,
  % ...
  language = {german}
}
```

For `babelbib` to take effect, a bibliography style supported by it - one of `babplain` , `babplai3` , `babalpha` , `babunsrt` , `bababbrv` , and `bababbr3` - must be used:

26 http://www.mps.mpg.de/software/latex/localtex/localltx.html#makebst
27 http://www.mps.mpg.de/software/latex/localtex/doc/merlin.pdf
28 http://tug.ctan.org/tex-archive/biblio/bibtex/contrib/babelbib/

```
\bibliographystyle{babplain}
\bibliography{sample}
```

38.3.11. Showing unused items

Usually LaTeX only displays the entries which are referred to with

```
\cite
```

. It's possible to make uncited entries visible:

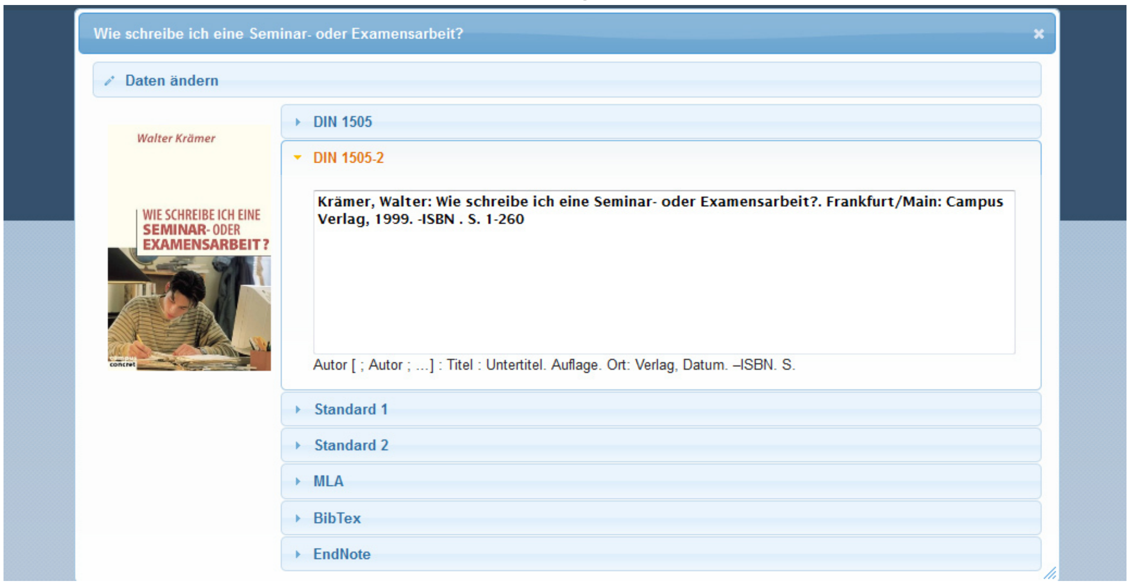
```
\nocite{Name89} % Show Bibliography entry of Name89
\nocite{*} % Show all Bib-entries
```

38.3.12. Getting bibliographic data

Many online databases provide bibliographic data in BibTeX-Format, making it easy to build your own database. For example, Google Scholar[29] offers the option to return properly formatted output, which can also be turned on in the settings page.
One should be alert to the fact that bibliographic databases are frequently the product of several generations of automatic processing, and so the resulting BibTex code is prone to a variety of minor errors, especially in older entries.

38.3.13. Helpful tools

See also: w:en:Comparison of reference management software[30]

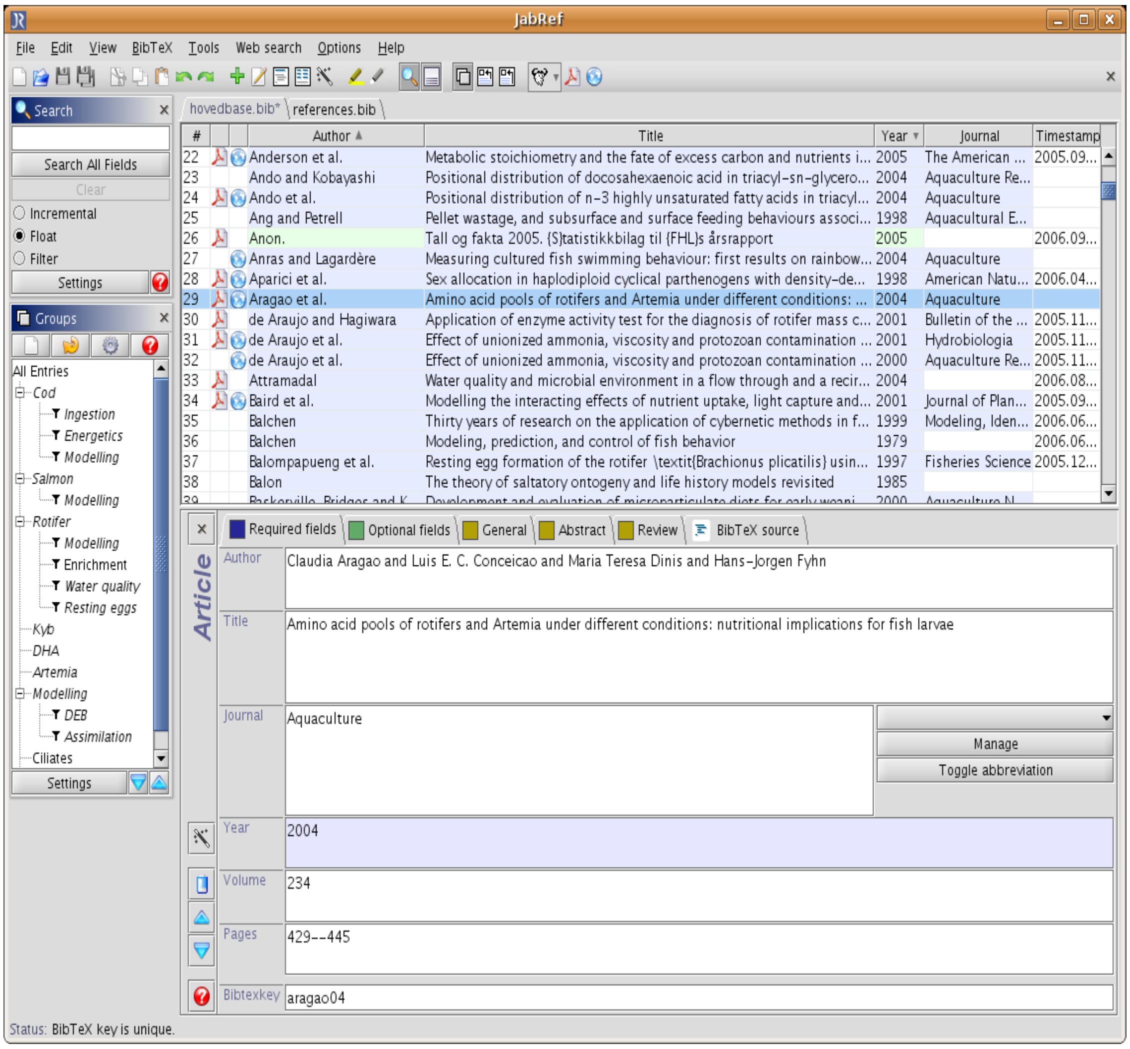

Wie schreibe ich eine Seminar- oder Examensarbeit?
Daten ändern
Walter Krämer
WIE SCHREIBE ICH EINE SEMINAR- ODER EXAMENSARBEIT?
DIN 1505
DIN 1505-2
Standard 1
Standard 2
MLA
BibTex
EndNote

Krämer, Walter: Wie schreibe ich eine Seminar- oder Examensarbeit?. Frankfurt/Main: Campus Verlag, 1999. -ISBN . S. 1-260

Autor [; Autor ; ...] : Titel : Untertitel. Auflage. Ort: Verlag, Datum. –ISBN. S.

Figure 150 Literatur-Generator

29 http://scholar.google.com
30 http://en.wikipedia.org/wiki/Comparison%20of%20reference%20management%20software

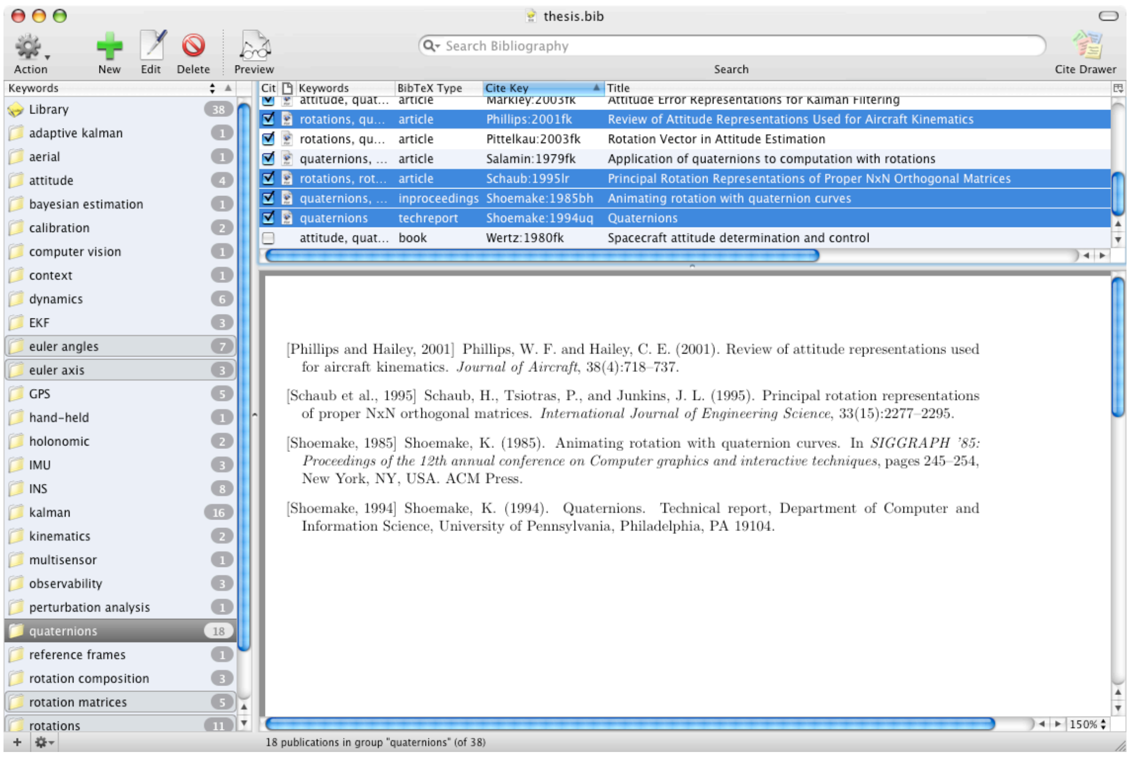


#	Author	Title	Year	Journal	Timestamp
22	Anderson et al.	Metabolic stoichiometry and the fate of excess carbon and nutrients i...	2005	The American ...	2005.09...
23	Ando and Kobayashi	Positional distribution of docosahexaenoic acid in triacyl-sn-glycero...	2004	Aquaculture Re...	
24	Ando et al.	Positional distribution of n-3 highly unsaturated fatty acids in triacyl...	2004	Aquaculture	
25	Ang and Petrell	Pellet wastage, and subsurface and surface feeding behaviours associ...	1998	Aquacultural E...	
26	Anon.	Tall og fakta 2005. {S}tatistikkbilag til {FHL}s årsrapport	2005		2006.09...
27	Anras and Lagardère	Measuring cultured fish swimming behaviour: first results on rainbow...	2004	Aquaculture	
28	Aparici et al.	Sex allocation in haplodiploid cyclical parthenogens with density-de...	1998	American Natu...	2006.04...
29	Aragao et al.	Amino acid pools of rotifers and Artemia under different conditions: ...	2004	Aquaculture	
30	de Araujo and Hagiwara	Application of enzyme activity test for the diagnosis of rotifer mass c...	2001	Bulletin of the ...	2005.11...
31	de Araujo et al.	Effect of unionized ammonia, viscosity and protozoan contamination ...	2001	Hydrobiologia	2005.11...
32	de Araujo et al.	Effect of unionized ammonia, viscosity and protozoan contamination ...	2000	Aquaculture Re...	2005.11...
33	Attramadal	Water quality and microbial environment in a flow through and a recir...	2004		2006.08...
34	Baird et al.	Modelling the interacting effects of nutrient uptake, light capture and...	2001	Journal of Plan...	2005.09...
35	Balchen	Thirty years of research on the application of cybernetic methods in f...	1999	Modeling, Iden...	2006.06...
36	Balchen	Modeling, prediction, and control of fish behavior	1979		2006.06...
37	Balompapueng et al.	Resting egg formation of the rotifer \textit{Brachionus plicatilis} usin...	1997	Fisheries Science	2005.12...
38	Balon	The theory of saltatory ontogeny and life history models revisited	1985		

Author	Claudia Aragao and Luis E. C. Conceicao and Maria Teresa Dinis and Hans-Jorgen Fyhn
Title	Amino acid pools of rotifers and Artemia under different conditions: nutritional implications for fish larvae
Journal	Aquaculture
Year	2004
Volume	234
Pages	429--445
Bibtexkey	aragao04

Status: BibTeX key is unique.

Figure 151 JabRef



Keywords	
Library	38
adaptive kalman	1
aerial	1
attitude	4
bayesian estimation	1
calibration	2
computer vision	1
context	1
dynamics	6
EKF	3
euler angles	7
euler axis	3
GPS	5
hand-held	1
holonomic	2
IMU	3
INS	8
kalman	16
kinematics	2
multisensor	1
observability	3
perturbation analysis	1
quaternions	18
reference frames	1
rotation composition	3
rotation matrices	5
rotations	11

Cit	Keywords	BibTeX Type	Cite Key	Title
✓	attitude, quat…	article	Markley:2003fk	Attitude Error Representations for Kalman Filtering
✓	rotations, qu…	article	Phillips:2001fk	Review of Attitude Representations Used for Aircraft Kinematics
✓	rotations, qu…	article	Pittelkau:2003fk	Rotation Vector in Attitude Estimation
✓	quaternions, …	article	Salamin:1979fk	Application of quaternions to computation with rotations
✓	rotations, rot…	article	Schaub:1995lr	Principal Rotation Representations of Proper NxN Orthogonal Matrices
✓	quaternions, …	inproceedings	Shoemake:1985bh	Animating rotation with quaternion curves
✓	quaternions	techreport	Shoemake:1994uq	Quaternions
☐	attitude, quat…	book	Wertz:1980fk	Spacecraft attitude determination and control

[Phillips and Hailey, 2001] Phillips, W. F. and Hailey, C. E. (2001). Review of attitude representations used for aircraft kinematics. *Journal of Aircraft*, 38(4):718–737.

[Schaub et al., 1995] Schaub, H., Tsiotras, P., and Junkins, J. L. (1995). Principal rotation representations of proper NxN orthogonal matrices. *International Journal of Engineering Science*, 33(15):2277–2295.

[Shoemake, 1985] Shoemake, K. (1985). Animating rotation with quaternion curves. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, New York, NY, USA. ACM Press.

[Shoemake, 1994] Shoemake, K. (1994). Quaternions. Technical report, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104.

150%

18 publications in group "quaternions" (of 38)

Figure 152 BibDesk

- BibDesk[31] BibDesk is a bibliographic reference manager for Mac OS X. It features a very usable user interface and provides a number of features like smart folders based on keywords and live tex display.
- BibSonomy[32] — A free social bookmark and publication management system based on BibTeX.
- BibTeXSearch[33] BibTeXSearch is a free searchable BibTeX database spanning millions of academic records.
- Bibtex Editor[34] - An online BibTeX entry generator and bibliography management system. Possible to import and export Bibtex files.
- Bibwiki[35] Bibwiki is a Specialpage for MediaWiki to manage BibTeX bibliographies. It offers a straightforward way to import and export bibliographic records.
- cb2Bib[36] The cb2Bib is a tool for rapidly extracting unformatted, or unstandardized bibliographic references from email alerts, journal Web pages, and PDF files.
- Citavi[37] Commercial software (with size-limited free demo version) which even searches libraries for citations and keeps all your knowledge in a database. Export of the database to all kinds of formats is possible. Works together with MS Word and Open

31 `http://bibdesk.sourceforge.net/`
32 `http://www.bibsonomy.org/`
33 `http://www.bibtexsearch.com/`
34 `http://truben.no/latex/bibtex`
35 `http://www.mediawiki.org/wiki/Extension:Bibwiki`
36 `http://www.molspaces.com/cb2bib/`
37 `http://www.citavi.ch`

Office Writer. Moreover plug ins for browsers and Acrobat Reader exist to automatically include references to your project.

- CiteULike[38] CiteULike is a free online service to organise academic papers. It can export citations in BibTeX format, and can "scrape" BibTeX data from many popular websites.
- DokuWiki[39] Bibtex is a DokuWiki plugin that allows for the inclusion of bibtex formatted citations in DokuWiki pages and displays them in APA format. Note: This Plugins is vulnerable to an XSS attack -> `http://www.dokuwiki.org/plugin:bibtex`
- Ebib[40] — a BibTeX database manager for Emacs[41], well resolved and never more than a few keystrokes away.
- JabRef[42] is a Java program (under the GPL license) which lets you search many bibliographic databases such as Medline, Citeseer, IEEEXplore and arXiv and feed and manage your BibTeX local databases with your selected articles. Based on BiBTeX, JabRef can export in many other output formats such as html, MS Word or EndNote. It can be used online without being installed (`http://jabref.sourceforge.net/jws/jabref.jnlp`)
- KBib[43] Another BibTeX editor for KDE. It has similar capabilities, and slightly different UI. Features include BibTeX reference generation from PDF files, plain text, DOI, arXiv & PubMed IDs. Web queries to Google Scholar, PubMer, arXiv and a number of other services are also supported.
- KBibTeX[44] KBibTeX is a BibTeX editor for KDE to edit bibliographies used with LaTeX. Features include comfortable input masks, starting web queries (e. g. Google or PubMed) and exporting to PDF, PostScript, RTF and XML/HTML. As KBibTeX is using KDE's KParts technology, KBibTeX can be embedded into Kile or Konqueror.
- Literatur-Generator[45] is a German-language online tool for creating a bibliography (Bibtex, Endnote, Din 1505, ...).
- Mendeley[46] Mendeley is cost-free academic software for managing PDFs which can manage a bibliography in Open Office and read BibTeX.
- Qiqqa[47] Qiqqa is a free research manager that has built-in support for automatically associating BibTeX records with your PDFs and a 'BibTeX Sniffer' for helping you semi-automatically find BibTeX records.
- Referencer[48] Referencer is a Gnome application to organise documents or references, and ultimately generate a BibTeX bibliography file.
- Synapsen[49] — Hypertextual Card Index / Reference Manager with special support for BiBTeX / biblatex, written in Java.

38 `http://www.citeulike.org/`
39 `http://stat.genopole.cnrs.fr/~cambroise/doku.php?id=softwares:dokuwikibibtexplugin`
40 `http://ebib.sourceforge.net/`
41 `http://en.wikipedia.org/wiki/Emacs`
42 `http://jabref.sourceforge.net/`
43 `http://users.tpg.com.au/thachly/kbib/`
44 `http://home.gna.org/kbibtex/`
45 `http://literatur-generator.de/`
46 `http://mendeley.com`
47 `http://www.qiqqa.com/`
48 `http://icculus.org/referencer/index.html`
49 `http://www.verzetteln.de/synapsen/`

- Zotero[50] Zotero is a free and open reference manager working as a Firefox plugin or standalone application, capable of importing and exporting bib files.

38.3.14. Summary

Although it can take a little time to get to grips with BibTeX, in the long term, it's an efficient way to handle your references. It's not uncommon to find .bib files on websites that people compile as a list of their own publications, or a survey of relevant works within a given topic, etc. Or in those huge, online bibliography databases, you often find BibTeX versions of publications, so it's a quick cut-and-paste into your own .bib file, and then no more hassle!
Having all your references in one place can be a big advantage. And having them in a structured form, that allows customizable output is another one. There are a variety of free utilities that can load your .bib files, and allow you to view them in a more efficient manner, as well as sort them and check for errors.

38.4. Bibliography in the table of contents

If you are writing a *book* or *report* , you'll likely insert your bibliography using something like:

```
\begin{thebibliography}{99}
\bibitem{bib:one_book} some information
\bibitem{bib:one_article} other information
\end{thebibliography}
```

Or, if you are using BibTeX, your references will be saved in a .bib file, and your TeX document will include the bibliography by these commands:

```
\bibliographystyle{plain}
\bibliography{mybibtexfile}
```

Both of these examples will create a chapter-like (or section-like) output showing all your references. But even though the resulting “References” looks like a chapter or section, it will not be handled quite the same: it will not appear in the Table of Contents.

38.4.1. Using tocbibind

The most comfortable way of adding your bibliography to the table of contents is to use the dedicated package **tocbibind** that works with many standard document classes. Simply include this code in the preamble of your document:

```
\usepackage[nottoc]{tocbibind}
```

This will include the Bibliography in the Table of Contents without numbering. If you want to have proper numbering, include the following code in the preamble:

```
\usepackage[nottoc,numbib]{tocbibind}
```

The **tocbibind** package can also handle including the List of Figures, List of Tables and the Table of Contents itself in the Table of Contents. It has many options for numbering, document structure etc. to fit almost any scenario. See the **tocbibind** CTAN page[51] for detailed documentation.

50 http://www.zotero.org/
51 http://www.ctan.org/tex-archive/macros/latex/contrib/tocbibind

38.4.2. Other methods

As unnumbered item

If you want your bibliography to be in the table of contents, just add the following two lines just before the *thebibliography* environment:

```
\clearpage
\addcontentsline{toc}{chapter}{Bibliography}
```

(OR

```
\addcontentsline{toc}{section}{Bibliography}
```

if you're writing an *article*)
The first line just terminates the current paragraph and page. If you are writing a *book* , use `\cleardoublepage` to match the style used. The second line will add a line in the Table of Contents (first option, *toc*), it will be like the ones created by chapters (second option, *chapter*), and the third argument will be printed on the corresponding line in the Table of Contents; here *Bibliography* was chosen because it's the same text the *thebibliography* environment will automatically write when you use it, but you are free to write whatever you like. If you are using separate bib file, add these lines between `\bibliographystyle` and `\bibliography` .
If you use hyperref[52] package, you should also use `\phantomsection` command to enable hyperlinking from the table of contents to bibliography.

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{chapter}{Bibliography}
```

This trick is particularly useful when you have to insert the bibliography in the Table of Contents, but it can work for anything. When LaTeX finds the code above, it will record the info as described and the current page number, inserting a new line in the Contents page.

As numbered item

If you instead want bibliography to be numbered section or chapter, you'll likely use this way:

```
\cleardoublepage % This is needed if the book class is used, to place the anchor
 in the correct page,
                 % because the bibliography will start on its own page.
                 % Use \clearpage instead if the document class uses the
 "oneside" argument
\renewcommand*{\refname}{} % This will define heading of bibliography to be
 empty, so you can...
\section{Bibliography}     % ...place a normal section heading before the
 bibliography entries.

\begin{thebibliography}{99}
...
\end{thebibliography}
```

Another even easier solution is to use

```
\section
```

inside of the

```
\renewcommand
```

52 Chapter 20 on page 255

block:

```
\renewcommand{\refname}{\section{Sources}} % Using "Sources" as the title of the
 section
\begin{thebibliography}{99}
...
\end{thebibliography}
```

You may wish to use

```
\renewcommand*{\refname}{\vspace*{-1em}}
```

followed by

```
\vspace*{-1em}
```

to counteract the extra space the blank

```
\refname
```

inserts.
If you are using BibTeX, the

```
\bibliography
```

command, and the book or report class, you will need to redefine

```
\bibname
```

instead of

```
\refname
```

like so.

```
\renewcommand{\bibname}{\section{Sources}} % Redefine bibname
\bibliographystyle{IEEEtran}              % Set any options you want
\bibliography{your_bib_file_names}        % Build the bibliography
```

38.5. biblatex

As we said before, biblatex is widely considered the 'successor' of BibTeX. Intended as a full replacement for BibTeX, it is more configurable in its output and provides a multitude of new styles (for output) and fields (for the database) that can be used in a document. For now, refer to its comprehensive documentation on CTAN[53].

38.5.1. Entry and field types in .bib files

The following table shows most field types. Some field types are lists, either `lists of person names` , others are `literal lists` . A `date` can either be given in parts or full, some `keys` are necessary, page references are provided as `ranges` and certain special fields contain `verbatim` code. There are many kinds of `titles` .

Hierarchic entry types			
Base type	**Multi-volume**	**Standalone part thereof**	**Supplemental material therein**
`@book`	`@mvbook`	`@inbook` , `@bookinbook`	`@suppbook`
`@periodical`	—	`@article`	`@suppperiodical`
`@collection`	`@mvcollection`	`@incollection`	`@suppcollection`
`@reference`	`@mvreference`	`@inreference`	—

53 `http://www.ctan.org/pkg/biblatex`

Hierarchic entry types			
Base type	**Multi-volume**	**Standalone part thereof**	**Supplemental material therein**
`@proceedings`	`@mvproceedings`	`@inproceedings` , `@conference`	—

Entry types in `.bib` files known by biblatex and field types supported,
either required `+` , alternatively required `±` , optional `^` , not supported (empty) or forbidden `-` ;
some types have been shortened: dot '.' truncates entry and tilde '~' repeats last full entry

	article	**book**	**mv~**	**in~**	**~let**	**collect.**	**mv~**	**in~**	**manual**	**misc**	**online**	**patent**	**period.**	**proceed.**	**mv~**	**in~**	**report**	**thesis**	**unpub.**
`author` , `authortype`	+	+	+	+	±	−	−	+	±	±	±	+	−	−	−	+	+	+	+
`editor` , `editortype`	^	^	^	^	±	+	+	+	±	±	±		+	+	+	+			
`editor`*X* , `editor`*X*`type`	^	^	^	^		^	^	^					^						
`holder`												^							
`bookauthor`				^															
`annotator` , `commentator`	^	^	^	^		^	^	^											
`translator` , `origlanguage`	^	^	^	^		^	^	^											
`afterword` , `foreword` , `introduction`		^	^	^		^	^	^											
`title`	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
`titleaddon` , `subtitle`	^	^	^	^	^	^	^	^	^	^	^	^	ˇ	^	^	^	^	^	^

maintitle , mainsubtitle , maintitleaddon		ˆ	−	ˆ		ˆ	−	ˆ						ˆ	−	ˆ				
booktitle				+				+								+				
booksubtitle , booktitleaddon				ˆ				ˆ								ˆ				
journalsubtitle	ˆ																			
journaltitle	+																			
eventdate , eventtitle , eventtitleaddon , venue														ˆ	ˆ	ˆ				
date , year	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±	
month	ˆ									ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	
edition		ˆ	ˆ	ˆ		ˆ	ˆ	ˆ												
issue , issuetitle , issuesubtitle	ˆ												ˆ							
number	ˆ	ˆ	ˆ	ˆ		ˆ	ˆ	ˆ				+	ˆ	ˆ	ˆ	ˆ	ˆ			

series	ˆ	ˆ	ˆ	ˆ		ˆ	ˆ	ˆ	ˆ				ˆ	ˆ	ˆ	ˆ			
chapter		ˆ	–	ˆ	ˆ	ˆ	–	ˆ	ˆ					ˆ	–	ˆ	ˆ	ˆ	
part		ˆ	–	ˆ		ˆ	–	ˆ						ˆ	–	ˆ			
volume	ˆ	ˆ	–	ˆ		ˆ	–	ˆ					ˆ	ˆ	–	ˆ			
volumes		ˆ	ˆ	ˆ		ˆ	ˆ	ˆ						ˆ	ˆ	ˆ			
version	ˆ								ˆ	ˆ	ˆ	ˆ					ˆ		
doi , eprint , eprintclass , eprinttype	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ		ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
eid	ˆ																		
isbn		ˆ	ˆ	ˆ		ˆ	ˆ	ˆ	ˆ					ˆ	ˆ	ˆ		ˆ	ˆ
isrn																	ˆ		
issn	ˆ												ˆ						
isan , ismn , iswc																			
url	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	+	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
urldate	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
location		ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ		ˆ		ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
publisher		ˆ	ˆ	ˆ		ˆ	ˆ	ˆ	ˆ					ˆ	ˆ	ˆ			
organization									ˆ	ˆ	ˆ			ˆ	ˆ	ˆ			
institution																	+	+	
type					ˆ				ˆ	ˆ		ˆ					+	+	
howpublished					ˆ					ˆ									ˆ
pages	ˆ	ˆ	–	ˆ	ˆ	ˆ	–	ˆ	ˆ					ˆ	–	ˆ	ˆ	ˆ	
pagetotal		ˆ	ˆ	–	ˆ	ˆ	ˆ	–	ˆ					ˆ	ˆ	–	ˆ	ˆ	

Some entry types are hard to distinguish and are treated the same by standard styles:
- `@article` is the same as hypothetic *`@inperiodical` and therefore encompasses existing `@suppperiodical`
- `@inbook` = `@bookinbook` = `@suppbook`
- `@collection` = `@reference`
- `@mvcollection` = `@mvreference`
- `@incollection` = `@suppcollection` = `@inreference`
- `@online` = `@electronic` = `@www`
- `@report` = `@techreport`
- `@thesis` = `@mastersthesis` = `@phdthesis`

Some field types are defined, but the documentation does not say which entry types they can be used with. This is either because they depend on another field being set to be useful or they can always be used in a user-defined manner, but will never be used in standard styles:
- `abstract` , `annotation`
- `entrysubtype`
- `file`
- `label`
- `library`
- `nameaddon`
- `origdate` , `origlocation` , `origpublisher`
- `origtitle` , `reprinttitle` , `indextitle`
- `pagination` , `bookpagination`
- `shortauthor` , `shorteditor` , `shorthand` , `shorthandintro` , `shortjournal` , `shortseries` `shorttitle`

The only field that is always mandatory, is `title` . All entry types also require either `date` or `year` and they specify which of `author` and `editor` they expect or whether they can use both. Some field types can optionally be used with any entry type:
- `addendum` , `note`
- `language`
- `pubstate`
- `urldate`

All physical (print) entry types share further optional field types:
- `url` , `doi`
- `eprint` , `eprintclass` , `eprinttype`

Multimedia entry types
- `@artwork`
- `@audio`
- `@image`
- `@movie`
- `@music`
- `@performance`
- `@video`
- `@software`

and legal entry types
- `@commentary`
- `@jurisdiction`
- `@legislation`

- `@legal`
- `@letter`
- `@review`
- `@standard`

are defined, but not yet supported (well).
The entry types `@bibnote` , `@set` and `@xdata` are special.

38.5.2. Printing bibliography

Presuming we have defined our references in a file called references.bib, we add this to biblatex by adding the following to the preamble:

```
\addbibresource{references.bib}
```

Print the bibiography with this macro (usually at the end of the document body):

```
\printbibliography
```

Printing separate bibliographies

We want to separate the bibliography into papers, books and others

```
\printbibliography[title={Book references},type=book]
\printbibliography[title={Article references},type=article]
\printbibliography[title={Other references}, nottype=article, nottype=book]
```

If the bib entries are located in multiple files we can add them like this:

```
\addbibresource{references.bib}
\addbibresource{other.bib}
```

We can also filter on other fields, such as entrysubtype. If we define our online resources like this:

```
@misc{some-resource,
   ...
   entrysubtype = {inet},
}
```

we filter with

```
\printbibliography[title={Online resources}, subtype=inet]
```

Example with prefix keys, subheadings and table of contents

As the numbering of the bibliographies are independent, it can be useful to also separate the bibliographies using prefixnumbers such as a, b and c. In addition we add a main heading for the bibliographies and add that to the table of contents.
To make Hyperref[54] links point to the correct bibliography section, we also add

```
\phantomsection
```

before printing each bibliography

```
\printbibheading[
heading=bibintoc, % bibintoc adds the Bibliography to the table of contents
title={Resources} % If we want to override the default title "Bibliography"
]
\phantomsection % Requires hyperref package
\printbibliography[title={Printed sources}, heading=subbibliography,
 prefixnumbers={a}, type=book, type=article]
```

54 Chapter 20.1 on page 255

```
\phantomsection
\printbibliography[title={Online resources}, heading=subbibliography,
 prefixnumbers={c}, subtype=inet]
\phantomsection
\printbibliography[title={Other}, heading=subbibliography, prefixnumbers={c},
 nottype=article, nottype=book, notsubtype=inet]
```

To add each of the bibliographies to the table of contents as sub-sections to the main Bibliography, replace `heading=subbibliography` with `heading=subbibintoc` .

38.6. Multiple bibliographies

38.6.1. Using `multibib`

This package is for multiple Bibliographies for different sections in your work. For example, you can generate a bibliography for each chapter. You can find information about the package on CTAN[55]

38.6.2. Using `bibtopic`

The bibtopic-Package[56] is created to differ the citations on more files, so that you can divide the bibliography into more parts.

```
\documentclass[11pt]{article}
\usepackage{bibtopic}
\begin{document}

\bibliographystyle{alpha}
\section{Testing}
Let's cite all the books: \cite{ColBenh:93} and
\cite{Munt:93}; and an article: \cite{RouxSmart:95}.

File books.bib is used for this listing:
\begin{btSect}{books}
 \section{References from books}
 \btPrintCited
\end{btSect}
Here, the articles.bib is used, and the listing is in plain-format instead of
 the standard alpha.
\begin{btSect}[plain]{articles}
 \section{References from articles}
 \btPrintCited
 \section{Articles not cited}
 \btPrintNotCited
\end{btSect}
Just print all entries here with \btPrintAll
\begin{btSect}[plain]{internet}
 \section{References from the internet}
 \btPrintAll
\end{btSect}
\end{document}
```

38.7. Notes and references

This page uses material from Andy Roberts' Getting to grips with LaTeX with permission from the author.

55 `http://ctan.org/pkg/multibib`
56 `http://ctan.org/pkg/bibtopic`

fr:LaTeX/Gestion de la bibliographie[57] ru:LaTeX/Управление библиографией[58]

57 http://fr.wikibooks.org/wiki/LaTeX%2FGestion%20de%20la%20bibliographie

58 http://ru.wikibooks.org/wiki/LaTeX%2F%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%B5%D0%B9

39. More Bibliographies

This is a gentle introduction to using some of the bibliography functionality available to LaTeX users beyond the BibTeX[1] basics. This introduction won't be discussing how to create new styles or packages but rather how to use some existing ones. It is worth noting that *Harvard* , for example, is a *citation* style. It is associated with an alphabetical reference list secondarily ordered on date, but the only strictly defined element of *Harvard* style is the citation in *author-date* format.

39.1. The example data

The database used for my examples contains just the following

```
@article{Erdos65,
        title = {Some very hard sums},
        journal={Difficult Maths Today},
        author={Paul Erd\H{o}s and Arend Heyting and Luitzen Egbertus Brouwer},
        year={1930},
        pages={30}
}
```

39.2. The limits of BibTeX styles

Using cite.sty and BibTeX makes it very easy to produce **some** bibliography styles. But *author-date* styles - for example the often mentioned, never defined "Harvard" - are not so easy. It's true that you can download some *.bst* files from CTAN that will handle some variants but using them is not always straightforward. This guide deals with *Natbib* a supplementary package that can access *.bib* files and has sophisticated functionality for producing custom or default author-year format citations and bibliographies as well as the numerical styles handled by BibTeX.

39.3. Natbib

Natbib is a package created by Patrick Daly as a replacement for the *cite.sty* package when *author-date* citation styles are required. Natbib provides three associated bibliography styles:

- plainnat
- abbrvnat
- unsrtnat

which correspond to the three styles available by default in BibTeX where you have a plain numbered style, an abbreviated numbered style and an unsorted numbered style. Alongside these new styles is an extended set of citation commands to provide flexible citation formats. These are

1 Chapter 38 on page 443

```
\citet[]{}
```
and

```
\citep[]{}
```
each of which has a number of variants.

39.3.1. The Preamble

All Natbib styles require that you load the package in your document preamble. So, a skeleton LaTeX file with Natbib might look like this:

```
\documentclass[]{article}
\usepackage[round]{natbib}

\begin{document}

Document body text with citations.

\bibliographystyle{plainnat}
\bibliography{myrefs}

\end{document}
```

Options

Options available with Natbib can be specified in the brackets on the \usepackage command. Among them are:

Option	**Effect**
round	()
square	[]
curly	{}
angle	<>
semicolon	separate citations with ***;***
colon	as semicolon
comma	separate with commas
authoryear	author-year citations
numbers	numeric citations
super	superscript citations
sort	multiple citations are ordered as in bibliography
sort&compress	as ***sort*** but number ranges are compressed and hyphenated
compress	number ranges are compressed and hyphenated but only where the 'natural' sort produces a continuous range
longnamesfirst	first citation is full author list and subsequent citations are abbreviated
sectionbib	allows multiple bibliographies in the same document
nonamebreak	forces all author names onto one line
merge	merges a citation with a previous citation
elide	elides any repeated elements in merged references
mcite	ignore merge

Clearly some of these options require explanation but that will be achieved via examples below. For now, we just note that they can be passed through *\usepackage[]{}* in the preamble of your LaTeX file.

39.4. Citation

39.4.1. Basic Citation Commands

To cite with Natbib, use the commands *\citet* or *\citep* in your document. The "plain" versions of these commands produced abbreviated lists in the case of multiple authors but both have *** variants which result in full author listings. We assume the use of the *round* option in these examples.

\citet and \citet*

The *\citet* command is used for *textual* citations, that is to say that author names appear in the text outside of the parenthetical reference to the date of publication. This command can take options for chapter, page numbers etc. Here are examples

\citet{Erdos65}	produces	Erdős et al. (1965)
\citet[chapter 2]{Erdos65}	produces	Erdős et al. (1965, chapter 2)
\citet[pp. 10-12]{Erdos65}	produces	Erdős et al. (1965, pp. 10-12)
\citet[see][chap. 2]{Erdos65}	produces	Erdős et al. (see 1965, chap. 2)

Here are the *\citet** versions

\citet*{Erdos65}	produces	Erdős, Heyting and Brouwer (1965)
\citet*[chapter 2]{Erdos65}	produces	Erdős , Heyting and Brouwer (1965, chapter 2)
\citet*[pp. 10-12]{Erdos65}	produces	Erdős , Heyting and Brouwer (1965, pp. 10-12)
\citet*[see][chap. 2]{Erdos65}	produces	Erdős , Heyting and Brouwer (see 1965, chap. 2)

\citep and \citep*

The *\citep* command is used where the author name is to appear inside the parentheses alongside the date.

\citep{Erdos65}	produces	(Erdős et al. 1965)
\citep[chapter 2]{Erdos65}	produces	(Erdős et al. 1965, chapter 2)
\citep[pp. 10-12]{Erdos65}	produces	(Erdős et al. 1965, pp. 10-12)
\citep[see][chap. 2]{Erdos65}	produces	(see Erdős et al., 1965, chap. 2)
\citep[e.g.][]{Erdos65}	produces	(e.g. Erdős et al., 1965)

Here are the *\citep** versions

\citep*{Erdos65}	produces	(Erdős, Heyting and Brouwer 1965)
\citep*[chapter 2]{Erdos65}	produces	(Erdős, Heyting and Brouwer 1965, chapter 2)

\citep*[pp. 10-12]{Erdos65}	produces	(Erdős , Heyting and Brouwer 1965, pp. 10-12)
\citep*[see][chap. 2]{Erdos65}	produces	(see Erdős , Heyting and Brouwer, 1965, chap. 2)
\citep*[e.g.][]{Erdos65}	produces	(e.g. Erdős , Heyting and Brouwer, 1965)

39.4.2. The Reference List

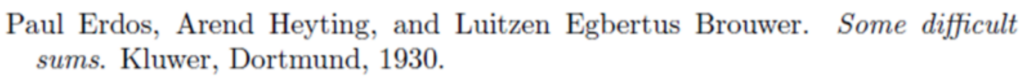
Having dealt with basic varieties of citation, we turn to the creation of the bibliography or reference list. Inserting a correct and correctly formatted bibliography when using Natbib is no different than when using plain BibTeX. There are two essential commands -

```
\bibliography{mybibliographydatabase}
```

which LaTeX interprets as an instruction to read a bibliographic database file (eg myrefs.bib) and insert the relevant data here, and

```
\bibliographystyle{plainnat}
```

which specifies how the data are to be presented. Above the three basic Natbib styles were mentioned as analogues of the partially homonymous styles in BibTeX. Let us imagine documents bearing citations as in the section about citation above[2]. Here is, approximately, how these citations would appear in plainnat.



Paul Erdos, Arend Heyting, and Luitzen Egbertus Brouwer. *Some difficult sums.* Kluwer, Dortmund, 1930.

Figure 153

39.4.3. What more is there?

This covers the basic functionality provided by the package Natbib. It may not, of course, provide what you are looking for. If you don't find what you want here then you should probably next investigate *harvard.sty* which provides a slighly different set of author-date citation functions. Providing a gentle guide to *harvard.sty* is my next rainy day project.

2 Chapter 39.4 on page 481

Part VI.

Special Documents

40. Letters

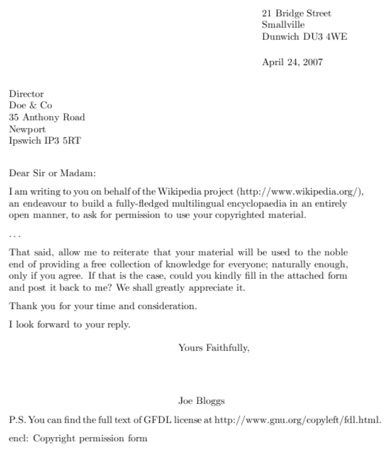
Sometimes the mundane things are the most painful. However, it doesn't have to be that way because of evolved, user-friendly templates. Thankfully, LaTeX allows for very quick letter writing, with little hassle.

40.1. The letter class

To write letters use the standard document class *letter* .
You can write multiple letters in one LaTeX file - start each one with `\begin{letter}{''recipient''}` and end with `\end{letter}`. You can leave *recipient* blank. Each letter consists of four parts.

1. Opening (like `\opening{Dear Sir or Madam,}` or `\opening{Dear Kate,}`).
2. Main body (written as usual in LaTeX). If you want the same body in all the letters, you may want to consider putting the entire body in a new command like `\newcommand{\BODY}{actual body}` and then using `\BODY` in all the letters.
3. Closing (like `\closing{Yours sincerely,}`).
   LaTeX will leave some space after closing for your hand-written signature; then it will put your name and surname, if you have declared them.
4. Additional elements: post scripta, carbon copy and list of enclosures.

If you want your name, address and telephone number to appear in the letter, you have to declare them first signature, address and telephone.
The output letter will look like this:



21 Bridge Street
Smallville
Dunwich DU3 4WE

April 24, 2007

Director
Doe & Co
35 Anthony Road
Newport
Ipswich IP3 5RT

Dear Sir or Madam:

I am writing to you on behalf of the Wikipedia project (http://www.wikipedia.org/), an endeavour to build a fully-fledged multilingual encyclopaedia in an entirely open manner, to ask for permission to use your copyrighted material.

. . .

That said, allow me to reiterate that your material will be used to the noble end of providing a free collection of knowledge for everyone; naturally enough, only if you agree. If that is the case, could you kindly fill in the attached form and post it back to me? We shall greatly appreciate it.

Thank you for your time and consideration.

I look forward to your reply.

Yours Faithfully,

Joe Bloggs

P.S. You can find the full text of GFDL license at http://www.gnu.org/copyleft/fdl.html.

encl: Copyright permission form

Figure 154 A sample letter.

Here is the example's code:

```
\documentclass{letter}
\usepackage{hyperref}
\signature{Joe Bloggs}
\address{21 Bridge Street \\ Smallville \\ Dunwich DU3 4WE}
\begin{document}

\begin{letter}{Director \\ Doe \& Co \\ 35 Anthony Road
\\ Newport \\ Ipswich IP3 5RT}
\opening{Dear Sir or Madam:}

I am writing to you on behalf of the Wikipedia project
 (http://www.wikipedia.org/),
an endeavour to build a fully-fledged multilingual encyclopaedia in an entirely
open manner, to ask for permission to use your copyrighted material.

% The \ldots command produces dots in a way that will not upset
% the typesetting of the document.
\ldots

That said, allow me to reiterate that your material will be used to the noble
 end of
providing a free collection of knowledge for everyone; naturally enough, only if
 you
agree. If that is the case, could you kindly fill in the attached form and post
 it
back to me? We shall greatly appreciate it.

Thank you for your time and consideration.

I look forward to your reply.

\closing{Yours Faithfully,}

\ps

P.S. You can find the full text of GFDL license at
\url{http://www.gnu.org/copyleft/fdl.html}.

\encl{Copyright permission form}

\end{letter}
\end{document}
```

To move the closing and signature parts to the left, insert the following before \begin{document}:

`\longindentation=0pt`

The amount of space to the left can be adjusted by increasing the 0pt.

40.2. Envelopes

40.2.1. Using the `envlab` package

The `envlab` package provides customization to the `\makelabels` command, allowing the user to print on any of an assortment of labels or envelope sizes. For example, beginning your LaTeX file the following way produces a document which includes the letter and a business-size (#10) envelope on the following page.

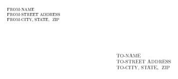
```
\documentclass{letter}
\usepackage[businessenvelope]{envlab}
\makelabels
```

Refer to the `envlab` user guide[1] for more information about this capable package. Note that the `envlab` package has issues displaying characters outside the base ASCII character set, see this bug report[2] for more information.

40.2.2. Using the `geometry` package

Here is a relatively simple envelope which uses the `geometry` package which is used because it vastly simplifies the task of rearranging things on the page (and the page itself).

```
% envelope.tex
\documentclass{letter}
\usepackage[
left=1in,top=0.15in,papersize={4.125in,9.5in},landscape,twoside=false]{geometry}
\setlength\parskip{0pt}
\pagestyle{empty}

\begin{document}

FROM-NAME

FROM-STREET ADDRESS

FROM-CITY, STATE, \ ZIP

\vspace{1.0in}\large
\setlength\parindent{3.6in}

TO-NAME

TO-STREET ADDRESS

TO-CITY, STATE, \ ZIP

\end{document}
```



FROM-NAME
FROM-STREET ADDRESS
FROM-CITY, STATE, ZIP

TO-NAME
TO-STREET ADDRESS
TO-CITY, STATE, ZIP

Figure 155 A sample envelope to be printed in landscape mode.

40.2.3. Printing

The above will certainly take care of the spacing but the actual printing is between you and your printer. One user reports that printing envelopes created with `envlab` is relatively painless. If you use the `geometry` package, you may find the following commands useful for printing the envelope.

1 `http://mirrors.ctan.org/macros/latex/contrib/envlab/elguide.pdf`
2 `http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=547978`

```
$ pdflatex envelope.tex
$ pdf2ps envelope.pdf
$ lpr -o landscape envelope.ps
```

Alternatively, you can use the latex dvi output driver.
In the first line, dvips command converts the .dvi file produced by latex into a .ps (PostScript) file. In the second line, the PostScript file is sent to the printer.

```
$ latex envelope.tex && dvips -t unknown -T 9.5in,4.125in envelope.dvi
$ lpr -o landscape envelope.ps
```

It is reported that `pdflatex` creates the right page size but not `dvips` despite what it says in the `geometry` manual. It will never work though unless your printer settings are adjusted to the correct page style. These settings depend on the printer filter you are using and in CUPS might be available on the `lpr` command line.

40.3. Windowed envelopes

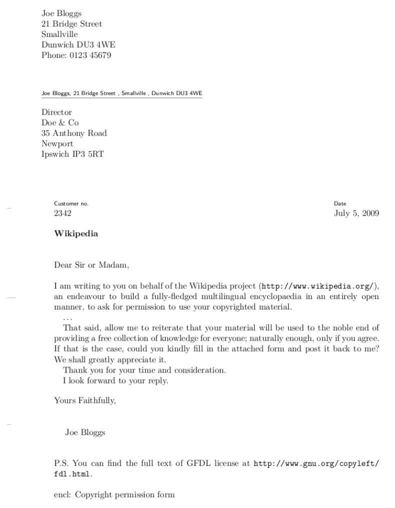
An alternative to separately printing addresses on envelopes is to use the letter class from the KOMA package. It supports additional features like folding marks and the correct address placement for windowed envelopes. Using the `scrlttr2` document class from the KOMA package the example letter code is:

```
% koma_env.tex
\documentclass[a4paper]{scrlttr2}
\usepackage{lmodern}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[english]{babel}
\usepackage{url}


\setkomavar{fromname}{Joe Bloggs}
\setkomavar{fromaddress}{21 Bridge Street \\ Smallville \\ Dunwich DU3 4WE}
\setkomavar{fromphone}{0123 45679}

\begin{document}

\begin{letter}{Director \\ Doe \& Co \\ 35 Anthony Road
\\ Newport \\ Ipswich IP3 5RT}

\KOMAoptions{fromphone=true,fromfax=false}
\setkomavar{subject}{Wikipedia}
\setkomavar{customer}{2342}
\opening{Dear Sir or Madam,}

I am writing to you on behalf of the Wikipedia project
(\url{http://www.wikipedia.org/}), an endeavour to build a
fully-fledged multilingual encyclopaedia in an entirely open
manner, to ask for permission to use your copyrighted material.

\ldots

That said, allow me to reiterate that your material will be used
to the noble end of providing a free collection of knowledge for
everyone; naturally enough, only if you agree. If that is the
case, could you kindly fill in the attached form and post it back
to me? We shall greatly appreciate it.
```

```
Thank you for your time and consideration.

I look forward to your reply.

\closing{Yours Faithfully,}
\ps{P.S. You can find the full text of GFDL license at
\url{http://www.gnu.org/copyleft/fdl.html}.}
\encl{Copyright permission form}

\end{letter}

\end{document}
```

The output is generated via

```
$ pdflatex koma_env
```



Joe Bloggs
21 Bridge Street
Smallville
Dunwich DU3 4WE
Phone: 0123 45679

Joe Bloggs, 21 Bridge Street , Smallville , Dunwich DU3 4WE

Director
Doe & Co
35 Anthony Road
Newport
Ipswich IP3 5RT

Customer no.	Date
2342	July 5, 2009

Wikipedia

Dear Sir or Madam,

I am writing to you on behalf of the Wikipedia project (`http://www.wikipedia.org/`), an endeavour to build a fully-fledged multilingual encyclopaedia in an entirely open manner, to ask for permission to use your copyrighted material.

...

That said, allow me to reiterate that your material will be used to the noble end of providing a free collection of knowledge for everyone; naturally enough, only if you agree. If that is the case, could you kindly fill in the attached form and post it back to me? We shall greatly appreciate it.

Thank you for your time and consideration.

I look forward to your reply.

Yours Faithfully,

Joe Bloggs

P.S. You can find the full text of GFDL license at `http://www.gnu.org/copyleft/fdl.html`.

encl: Copyright permission form

Figure 156 A sample letter with folding marks ready for standardized windowed envelopes.

Folding the print of the resulting file koma_env.pdf according the folding marks it can be placed into standardized windowed envelopes DIN C6/5, DL, C4, C5 or C6.
In addition to the default, the KOMA-package includes predefined format definitions for different standardized Swiss and Japanese letter formats.

40.4. Reference: `letter.cls` commands

command	**description**
\name{}	
\signature{}	
\address{}	
\location{}	

command	**description**
\telephone{}	
\makelabels	
\stopbreaks	
\startbreaks	
\opening{}	
\closing{}	
\cc{}	Start a parbox introduced with \ccname:
\encl{}	Start a parbox introduced with \enclname:
\ps	Begins a new paragraph, normally at the close of the letter
\stopletter	(empty)
\returnaddress	(empty)
\startlabels	
\mlabel{}{}	
\descriptionla-bel{}	
\ccname	”cc”
\enclname	”encl”
\pagename	”Page”
\headtoname	”To”
\date{}	Alter the date. See *datetime* package for alternative formattings.
\today	Long form date

environment	**Description**
letter{}	See main article
description	
verse	
quotation	
quote	

40.5. Sources

- KOMA-Script - The Guide[3]

pl:LaTeX/Pisanie listów[4] fr:LaTeX/Lettre[5]

3 `http://mirrors.ctan.org/macros/latex/contrib/koma-script/doc/scrguien.pdf`
4 `http://pl.wikibooks.org/wiki/LaTeX%2FPisanie%20list%C3%B3w`
5 `http://fr.wikibooks.org/wiki/LaTeX%2FLettre`

41. Presentations

LaTeX can be used for creating presentations. There are several packages for the task, including the `beamer` package.

41.1. The Beamer package

The beamer package is provided with most LaTeX distributions, but is also available from CTAN[1]. If you use MikTeX, all you have to do is to include the beamer package and let LaTeX download all wanted packages automatically. The documentation[2] explains the features in great detail. You can also have a look at the PracTex article **Beamer by example** .[3]
The `beamer` package also loads many useful packages including `hyperref` .

41.1.1. Introductory example

The beamer package is loaded by calling the `beamer` class:

```
\documentclass{beamer}
```

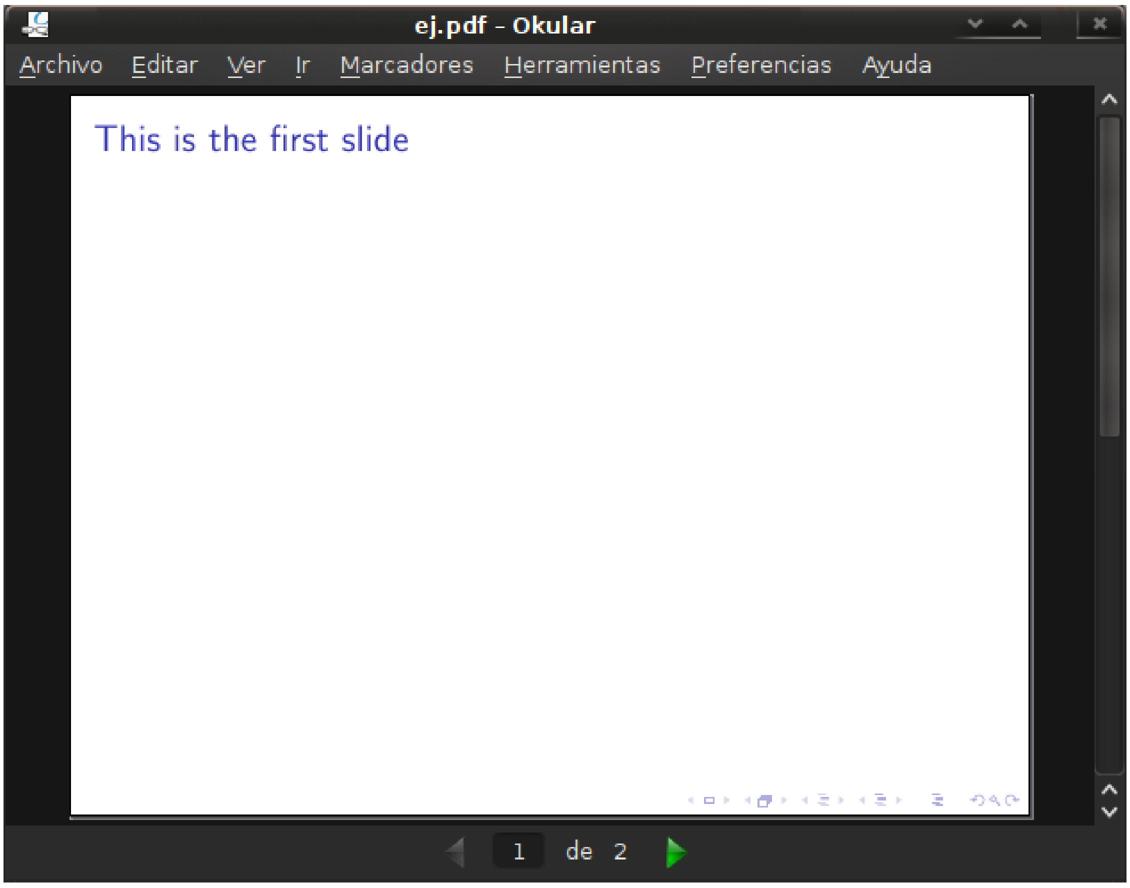
The usual header information may then be specified. Note that if you are compiling with XeTeX then you should use

```
\documentclass[xetex,mathserif,serif]{beamer}
```

Inside the `document` environment, multiple `frame` environments specify the content to be put on each slide. The `frametitle` command specifies the title for each slide (see image):

```
\begin{document}
  \begin{frame}
    \frametitle{This is the first slide}
    %Content goes here
  \end{frame}
  \begin{frame}
    \frametitle{This is the second slide}
    \framesubtitle{A bit more information about this}
    %More content goes here
  \end{frame}
% etc
\end{document}
```

1 `http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/`
2 `http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf`
3 Andrew Mertz and William Slough *Beamer by Example*


ej.pdf - Okular
Archivo Editar Ver Ir Marcadores Herramientas Preferencias Ayuda
1 de 2

This is the first slide

Figure 157

The usual environments (`itemize` , `enumerate` , `equation` , etc.) may be used.
Inside frames, you can use environments like `block` , `theorem` , `proof` , ... Also, `\maketitle` is possible to create the frontpage, if `title` and `author` are set.
Trick: Instead of using

```
\begin{frame}...\end{frame}
```

, you can also use

```
\frame{...}
```

.
For the actual talk, if you can compile it with `pdflatex` then you could use a pdf reader with a fullscreen mode, such as Okular[4], Evince[5] or Adobe Reader. If you want to navigate in your presentation, you can use the almost invisible links in the bottom right corner without leaving the fullscreen mode.

41.1.2. Document Structure

Title page and information

You give information about authors, titles and dates in the preamble.

4 https://en.wikipedia.org/wiki/Okular
5 https://en.wikipedia.org/wiki/Evince

```
\title[Crisis] % (optional, only for long titles)
{The Economics of Financial Crisis}
\subtitle{Evidence from India}
\author[Author, Anders] % (optional, for multiple authors)
{F.~Author\inst{1} \and S.~Anders\inst{2}}
\institute[Universities Here and There] % (optional)
{
  \inst{1}%
  Institute of Computer Science\\
  University Here
  \and
  \inst{2}%
  Institute of Theoretical Philosophy\\
  University There
}
\date[KPT 2004] % (optional)
{Conference on Presentation Techniques, 2004}
\subject{Computer Science}
```

In the document, you add the title page :

```
\frame{\titlepage}
```

Table of Contents

The table of contents, with the current section highlighted, is displayed by:

```
\begin{frame}
\frametitle{Table of Contents}
\tableofcontents[currentsection]
\end{frame}
```

This can be done automatically at the beginning of each section using the following code in the preamble:

```
\AtBeginSection[]
{
  \begin{frame}
    \frametitle{Table of Contents}
    \tableofcontents[currentsection]
  \end{frame}
}
```

Or for subsections:

```
\AtBeginSubsection[]
{
  \begin{frame}
    \frametitle{Table of Contents}
    \tableofcontents[currentsection,currentsubsection]
  \end{frame}
}
```

References (Beamer)

Beamer does not officially support BibTeX. Instead bibliography items will need to be partly set "by hand" (see beameruserguide.pdf 3.12[6]). The following example shows a references slide containing two entries:

```
\begin{frame}[allowframebreaks]
  \frametitle<presentation>{Further Reading}
  \begin{thebibliography}{10}
```

6 http://www.tex.ac.uk/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf

```
  \beamertemplatebookbibitems
  \bibitem{Autor1990}
    A.~Autor.
    \newblock {\em Introduction to Giving Presentations}.
    \newblock Klein-Verlag, 1990.
  \beamertemplatearticlebibitems
  \bibitem{Jemand2000}
    S.~Jemand.
    \newblock On this and that.
    \newblock {\em Journal of This and That}, 2(1):50--100, 2000.
  \end{thebibliography}
\end{frame}
```

As the reference list grows, the reference slide will divide into two slides and so on, through use of the `allowframebreaks` option. Individual items can be cited after adding an 'optional' label to the relevant `bibitem` stanza. The citation call is simply

```
\cite
```

. Beamer also supports limited customization of the way references are presented (see the manual). Those who wish to use natbib[7], for example, with Beamer may need to troubleshoot both their document setup and the relevant BibTeX style file.
The different types of referenced work are indicated with a little symbol (e.g. a book, an article, etc.). The Symbol is set with the commands `beamertemplatebookbibitems` and `beamertemplatearticlebibitems` . It is also possible to use `setbeamertemplate` directly, like so

```
\begin{frame}[allowframebreaks]
  \frametitle<presentation>{Further Reading}
  \begin{thebibliography}{10}
  \setbeamertemplate{bibliography item}[book]
  \bibitem{Autor1990}
    A.~Autor.
    \newblock {\em Introduction to Giving Presentations}.
    \newblock Klein-Verlag, 1990.
  \setbeamertemplate{bibliography item}[article]
  \bibitem{Jemand2000}
    S.~Jemand.
    \newblock On this and that.
    \newblock {\em Journal of This and That}, 2(1):50--100, 2000.
  \end{thebibliography}
\end{frame}
```

Other possible types of bibliography items, besides `book` and `article` , include e.g. `online` , `triangle` and `text` . It is also possible to have user defined bibliography items by including a graphic.

41.1.3. Style

Themes

The first solution is to use a built-in theme such as Warsaw, Berlin, etc. The second solution is to specify colors, inner themes and outer themes.

The Built-in solution

To the preamble you can add the following line:

```
\usetheme{Warsaw}
```

7 `http://www.ctan.org/pkg/natbib/`

to use the "Warsaw" theme. `Beamer` has several themes, many of which are named after cities (e.g. Frankfurt, Madrid, Berlin, etc.).
This Theme Matrix[8] contains the various theme and color combinations included with `beamer` . For more customizing options, have a look to the official documentation included in your distribution of `beamer` , particularly the part *Change the way it looks* .
The full list of themes is:

8 `http://www.hartwork.org/beamer-theme-matrix/`

• Antibes	• Darmstadt	• Ilmenau	• Marburg	• Singapore	• CambridgeUS
• Bergen	• Dresden	• JuanLesPins	• Montpellier	• Szeged	•
• Berkeley	• Frankfurt	• Luebeck	• PaloAlto	• Warsaw	•
• Berlin	• Goettingen	• Madrid	• Pittsburgh	• boxes	•
• Copenhagen	• Hannover	• Malmoe	• Rochester	• default	•

Color themes, typically with animal names, can be specified with

```
\usecolortheme{beaver}
```

The full list of color themes is:

- default
- albatross
- beaver
- beetle
- crane
- dolphin
- dove
- fly
- lily
- orchid
- rose
- seagull
- seahorse
- whale
- wolverine

The *do it yourself* solution

First you can specify the *outertheme* . The outertheme defines the head and the footline of each slide.

```
\useoutertheme{infolines}
```

Here is a list of all available outer themes:
- infolines
- miniframes
- shadow
- sidebar
- smoothbars
- smoothtree
- split
- tree

Then you can add the innertheme:

```
\useinnertheme{rectangles}
```

Here is a list of all available inner themes:
- rectangles
- circles
- inmargin
- rounded

You can define the color of every element:

```
\setbeamercolor{alerted text}{fg=orange}
\setbeamercolor{background canvas}{bg=white}
\setbeamercolor{block body alerted}{bg=normal text.bg!90!black}
\setbeamercolor{block body}{bg=normal text.bg!90!black}
\setbeamercolor{block body example}{bg=normal text.bg!90!black}
\setbeamercolor{block title alerted}{use={normal text,alerted text},fg=alerted
 text.fg!75!normal text.fg,bg=normal text.bg!75!black}
\setbeamercolor{block title}{bg=blue}
\setbeamercolor{block title example}{use={normal text,example text},fg=example
 text.fg!75!normal text.fg,bg=normal text.bg!75!black}
\setbeamercolor{fine separation line}{}
\setbeamercolor{frametitle}{fg=brown}
\setbeamercolor{item projected}{fg=black}
\setbeamercolor{normal text}{bg=black,fg=yellow}
\setbeamercolor{palette sidebar primary}{use=normal text,fg=normal text.fg}
\setbeamercolor{palette sidebar quaternary}{use=structure,fg=structure.fg}
\setbeamercolor{palette sidebar secondary}{use=structure,fg=structure.fg}
```

```
\setbeamercolor{palette sidebar tertiary}{use=normal text,fg=normal text.fg}
\setbeamercolor{section in sidebar}{fg=brown}
\setbeamercolor{section in sidebar shaded}{fg=grey}
\setbeamercolor{separation line}{}
\setbeamercolor{sidebar}{bg=red}
\setbeamercolor{sidebar}{parent=palette primary}
\setbeamercolor{structure}{bg=black, fg=green}
\setbeamercolor{subsection in sidebar}{fg=brown}
\setbeamercolor{subsection in sidebar shaded}{fg=grey}
\setbeamercolor{title}{fg=brown}
\setbeamercolor{titlelike}{fg=brown}
```

Colors can be defined as usual:

```
\definecolor{chocolate}{RGB}{33,33,33}
```

Block styles can also be defined:

```
\setbeamertemplate{blocks}[rounded][shadow=true]
\setbeamertemplate{background canvas}[vertical
 shading][bottom=white,top=structure.fg!25]
\setbeamertemplate{sidebar canvas left}[horizontal
 shading][left=white!40!black,right=black]
```

You can also suppress the navigation bar:

```
\beamertemplatenavigationsymbolsempty
```

Fonts

You may also change the fonts for particular elements. If you wanted the title of the presentation as rendered by

```
\frame{\titlepage}
```

to occur in a serif font instead of the default sanserif, you would use:

```
\setbeamerfont{title}{family=\rm}
```

You could take this a step further if you are using OpenType fonts with Xe(La)TeX and specify a serif font with increased size and oldstyle proportional alternate number glyphs:

```
\setbeamerfont{title}{family=\rm\addfontfeatures{Scale=1.18, Numbers={Lining,
 Proportional}}}
```

Math Fonts

The default settings for `beamer` use a different set of math fonts than one would expect from creating a simple math article. One quick fix for this is to use at the beginning of the file the option `mathserif`

```
\documentclass[mathserif]{beamer}
```

Others have proposed to use the command

```
\usefonttheme[onlymath]{serif}
```

but it is not clear if this works for absolutely every math character.

41.1.4. Frames Options

The *plain* option. Sometimes you need to include a large figure or a large table and you don't want to have the bottom and the top off the slides. In that case, use the plain option:

```
\frame[plain]{
```

```
% ...
}
```

If you want to include lots of text on a slide, use the *shrink* option.

```
\frame[shrink]{
% ...
}
```

The *allowframebreaks* option will auto-create new frames if there is too much content to be displayed on one.

```
\frame[allowframebreaks]{
% ...
}
```

Before using any verbatim environment (like `listings`), you should pass the option `fragile` to the

`frame`

environment, as verbatim environments need to be typeset differently. Usually, the form `fragile=singleslide` is usable (for details see the manual). Note that the `fragile` option may not be used with

`\frame`

commands since it expects to encounter a

`\end{frame}`

, which should be alone on a single line.

```
\begin{frame}[fragile]
\frametitle{Source code}

\begin{lstlisting}[caption=First C example]
int main()
{
    printf("Hello World!");
    return 0;
}
\end{lstlisting}
\end{frame}
```

41.1.5. Hyperlink navigation

Internal and external hyperlinks can be used in beamer to assist navigation. Clean looking buttons can also be added.
By default the beamer class adds navigation buttons in the bottom right corner. To remove them one can place

`\beamertemplatenavigationsymbolsempty`

in the preamble.

41.1.6. Animations

The following is merely an introduction to the possibilities in beamer. Chapter 8 of the beamer manual provides much more detail, on many more features.
Making items appear on a slide is possible by simply using the

`\pause`

statement:

```
\begin{frame}
\frametitle{Some background}
```

```
We start our discussion with some concepts.
\pause
The first concept we introduce originates with Erd\H os.
\end{frame}
```

Text or figures after

```
\pause
```

will display after one of the following events (which may vary between PDF viewers): pressing space, return or page down on the keyboard, or using the mouse to scroll down or click the next slide button. Pause can be used within

```
\itemize
```

etc.

Text animations

For text animations, for example in the itemize environment, it is possible to specify appearance and disappearance of text by using

```
<a-b>
```

where **a** and **b** are the numbers of the events the item is to be displayed for (inclusive). For example:

```
\begin{itemize}
  \item This one is always shown
  \item<1-> The first time (i.e. as soon as the slide loads)
  \item<2-> The second time
  \item<1-> Also the first time
  \only<1-1> {This one is shown at the first time, but it will hide soon (on the
 next event after the slide loads).}
\end{itemize}
```

A simpler approach for revealing one item per click is to use

```
\begin{itemize}[<+->]
```

.

```
\begin{frame}
        \frametitle{`Hidden higher-order concepts?'}
        \begin{itemize}[<+->]
        \item The truths of arithmetic which are independent of PA in some
        sense themselves `{contain} essentially {\color{blue}{hidden higher-order}},
         or infinitary, concepts'???
        \item `Truths in the language of arithmetic which \ldots
        \item       That suggests stronger version of Isaacson's thesis.
        \end{itemize}
\end{frame}
```

In all these cases, pressing page up, scrolling up, or clicking the previous slide button in the navigation bar will backtrack through the sequence.

41.1.7. Handout mode

In beamer class, the default mode is *presentation* which makes the slides. However, you can work in a different mode that is called *handout* by setting this option when calling the class:

```
\documentclass[12pt,handout]{beamer}
```

This mode is useful to see each slide only one time with all its stuff on it, making any

```
\itemize[<+->]
```

environments visible all at once (for instance, printable version). Nevertheless, this makes an issue when working with the

```
only
```

command, because its purpose is to have *only* some text or figures at a time and not all of them together.
If you want to solve this, you can add a statement to specify precisely the behavior when dealing with

```
only
```

commands in handout mode. Suppose you have a code like this

```
\only<1>{\includegraphics{pic1.eps}}
\only<2>{\includegraphics{pic2.eps}}
```

These pictures being completely different, you want them both in the handout, but they cannot be both on the same slide since they are large. The solution is to add the handout statement to have the following:

```
\only<1| handout:1>{\includegraphics{pic1.eps}}
\only<2| handout:2>{\includegraphics{pic2.eps}}
```

This will ensure the handout will make a slide for each picture.
Now imagine you still have your two pictures with the only statements, but the second one show the first one plus some other graphs and you don't need the first one to appear in the handout. You can thus precise the handout mode not to include some only commands by:

```
\only<1| handout:0>{\includegraphics{pic1.eps}}
\only<2>{\includegraphics{pic2.eps}}
```

The command can also be used to hide frames, e.g.

```
\begin{frame}<handout:0>
```

or even, if you have written a frame that you don't want anymore but maybe you will need it later, you can write

```
\begin{frame}<0| handout:0>
```

and this will hide your slide in both modes. (The order matters. Don't put handout:0|beamer:0 or it won't work.)
A last word about the handout mode is about the notes. Actually, the full syntax for a frame is

```
\begin{frame}
...
\end{frame}
\note{...}
\note{...}
...
```

and you can write your notes about a frame in the field *note* (many of them if needed). Using this, you can add an option to the class calling, either

```
\documentclass[12pt,handout,notes=only]{beamer}
```

or

```
\documentclass[12pt,handout,notes=show]{beamer}
```

The first one is useful when you make a presentation to have only the notes you need, while the second one could be given to those who have followed your presentation or those who missed it, for them to have both the slides with what you said.
Note that the 'handout' option in the \documentclass line suppress all the animations.

Important: the *notes=only* mode is **literally** doing only the notes. This means there will be no output file but the DVI. Thus it requires you to have run the compilation in another mode before. If you use separate files for a better distinction between the modes, you may need to copy the .aux file from the handout compilation with the slides (w/o the notes).

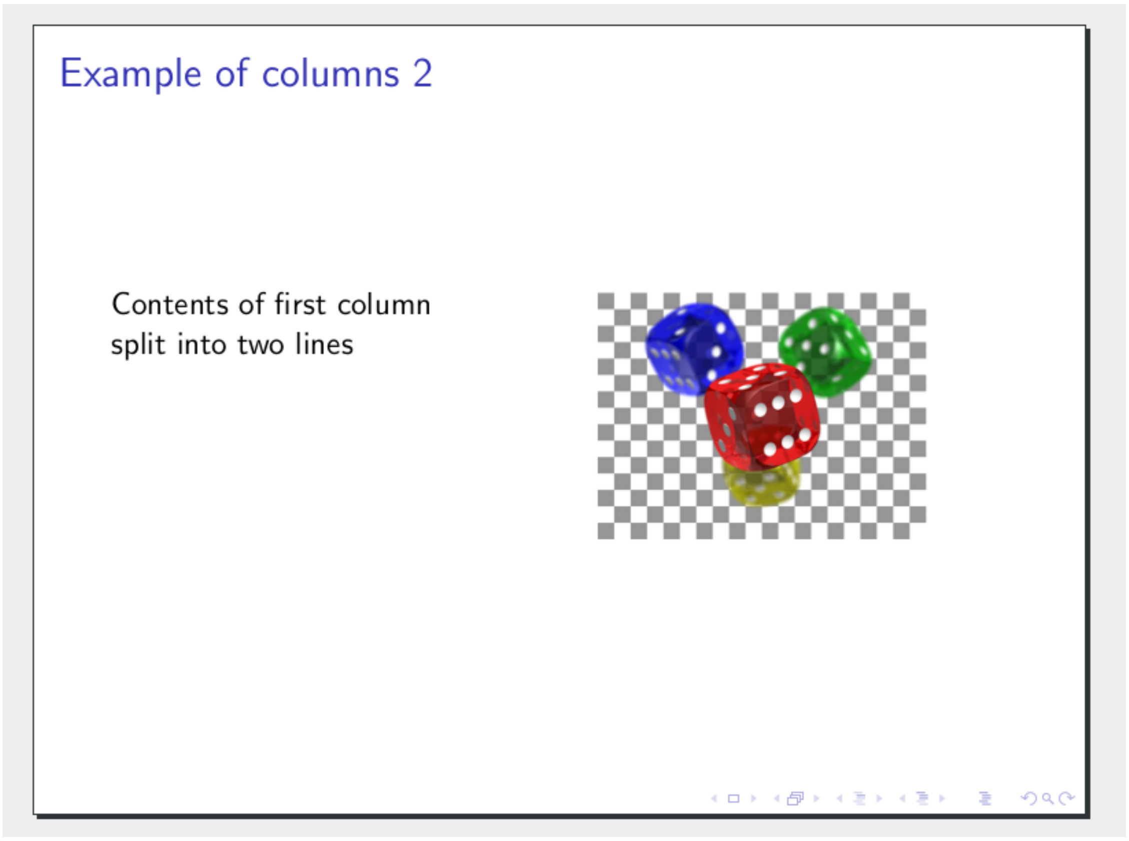
41.1.8. Columns and Blocks

There are two handy environments for structuring a slide: ”blocks”, which divide the slide (horizontally) into headed sections, and ”columns” which divides a slide (vertically) into columns. Blocks and columns can be used inside each other.

Columns

Example

```
\begin{frame}{Example of columns 1}
    \begin{columns}[c] % the "c" option specifies center vertical alignment
    \column{.5\textwidth} % column designated by a command
     Contents of the first column
    \column{.5\textwidth}
     Contents split \\ into two lines
    \end{columns}
\end{frame}

\begin{frame}{Example of columns 2}
     \begin{columns}[T] % contents are top vertically aligned
     \begin{column}[T]{5cm} % each column can also be its own environment
     Contents of first column \\ split into two lines
     \end{column}
     \begin{column}[T]{5cm} % alternative top-align that's better for graphics
          \includegraphics[height=3cm]{graphic.png}
     \end{column}
     \end{columns}
\end{frame}
```



Example of columns 2

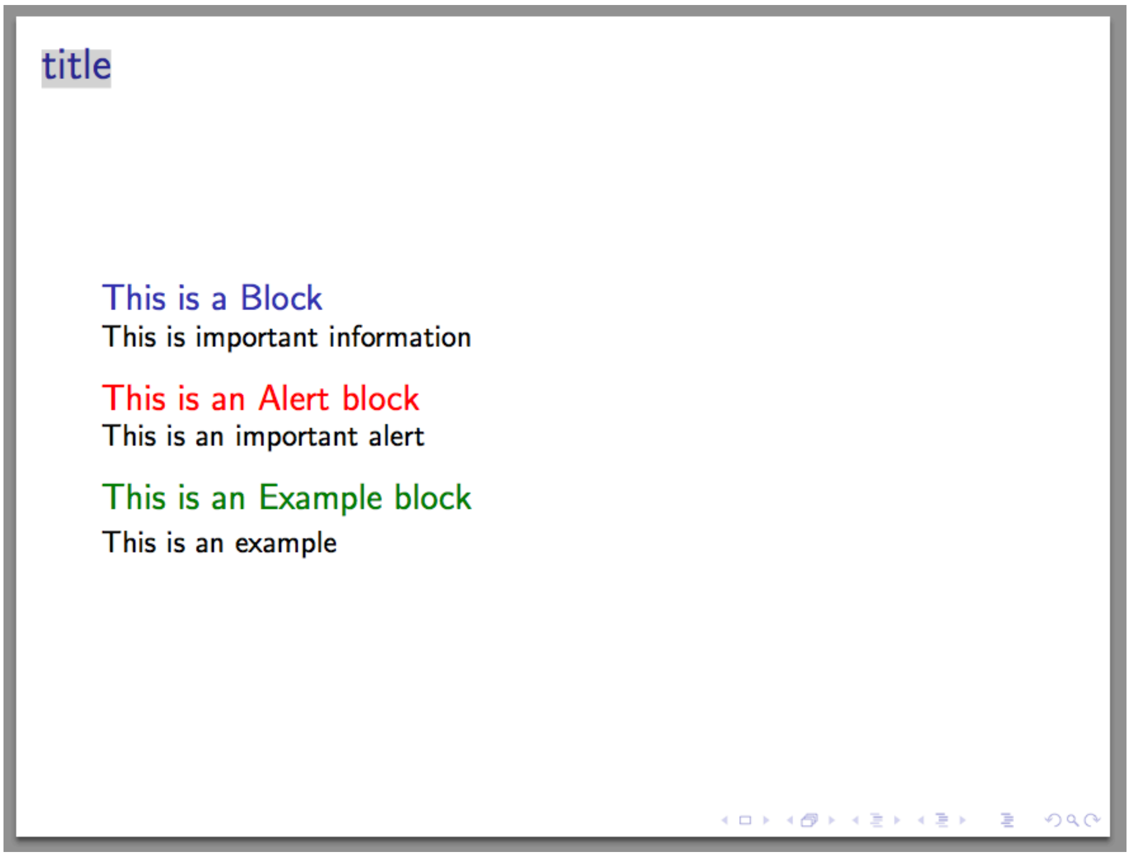
Contents of first column
split into two lines

Figure 158 Example of columns in Beamer

Blocks

Enclosing text in the *block* environment creates a distinct, headed block of text (a blank heading can be used). This allows to visually distinguish parts of a slide easily. There are three basic types of block. Their formatting depends on the theme being used.
Simple

```
\begin{frame}

   \begin{block}{This is a Block}
      This is important information
   \end{block}

   \begin{alertblock}{This is an Alert block}
   This is an important alert
   \end{alertblock}

   \begin{exampleblock}{This is an Example block}
   This is an example
   \end{exampleblock}

\end{frame}
```


title

This is a Block

This is important information

This is an Alert block

This is an important alert

This is an Example block

This is an example

Figure 159 Ejemplo de bloques en una presentación con Beamer

41.1.9. PDF options

You can specify the default options of your PDF.[9]

```
\hypersetup{pdfstartview={Fit}} % fits the presentation to the window when first
 displayed
```

41.2. The powerdot package

The powerdot package is an alternative to beamer. It is available from CTAN[10]. The documentation[11] explains the features in great detail.
The powerdot package is loaded by calling the `powerdot` class:

```
\documentclass{powerdot}
```

The usual header information may then be specified.
Inside the usual `document` environment, multiple `slide` environments specify the content to be put on each slide.

9 Other possible values are defined in the hyperref manual ^{`http://mirror.switch.ch/ftp/mirror/tex/macros/latex/contrib/hyperref/doc/manual.html#TBL-7-40-1`}

10 `http://www.ctan.org/tex-archive/macros/latex/contrib/powerdot/`

11 `http://mirrors.ctan.org/macros/latex/contrib/powerdot/doc/powerdot.pdf`

```
\begin{document}
  \begin{slide}{This is the first slide}
    %Content goes here
  \end{slide}
  \begin{slide}{This is the second slide}
    %More content goes here
  \end{slide}
% etc
\end{document}
```

41.3. References

41.4. Links

- Wikipedia:Beamer (LaTeX)[12]
- Beamer user guide[13] (pdf) from CTAN
- The powerdot class[14] (pdf) from CTAN
- A tutorial for creating presentations using beamer[15]

fr:LaTeX/Faire des présentations[16]

12 http://en.wikipedia.org/wiki/Beamer%20%28LaTeX%29
13 http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf
14 http://mirrors.ctan.org/macros/latex/contrib/powerdot/doc/powerdot.pdf
15 http://www.math-linux.com/spip.php?article77
16 http://fr.wikibooks.org/wiki/LaTeX%2FFaire%20des%20pr%C3%A9sentations

42. Teacher's Corner

42.1. Intro

LaTeX has specific features for teachers. We present the **exam** class[1] which is useful for designing exams and exercises with solutions. Interested people could also have a look at the **probsoln** package[2], the **mathexm** document class[3], or the **exsheets** package[4].

42.2. The exam class

We present the **exam** class. The exam class is well suited to design exams with solutions. You just have to specify in the preamble if you want the solutions to be printed or not. You can also count the number of points.

42.2.1. Preamble

In the preamble you can specify the following lines :

```
\documentclass[a4paper,11pt]{exam}
\printanswers % If you want to print answers
% \noprintanswers % If you don't want to print answers
\addpoints % if you want to count the points
% \noaddpoints % if you don't want to count the points
% Specifies the way question are displayed:
\qformat{\textbf{Question\thequestion}\quad(\thepoints)\hfill}
\usepackage{color} % defines a new color
\definecolor{SolutionColor}{rgb}{0.8,0.9,1} % light blue
\shadedsolutions % defines the style of the solution environment
% \framedsolutions % defines the style of the solution environment
% Defines the title of the solution environment:
\renewcommand{\solutiontitle}{\noindent\textbf{Solution:}\par\noindent}
```

You can replace the 3 first lines with the following :

```
\documentclass[a4paper,11pt,answers,addpoints]{exam}
```

42.2.2. Document

- The exam is included in the **questions** environment.
- The command \question introduces a new question.
- The number of points is specified in squared brackets.
- The solution is given in the **solution** environment. It appears only if \printanswers or **answers** as an option of the \documentclass are specified in the preamble.

1 examdoc ^{`http://www-math.mit.edu/~psh/exam/examdoc.pdf`} Using the exam document class
2 Probsoln ^{`http://www.tex.ac.uk/tex-archive/macros/latex/contrib/probsoln/probsoln.pdf`} Creating problem sheets optionally with solutions
3 mathexm documentation ^{`http://mirrors.ctan.org/macros/latex/contrib/mathexam/doc/mathexam.pdf`}
4 exsheets documentation ^{`http://mirrors.ctan.org/macros/latex/contrib/exsheets/exsheets_en.pdf`} Create exercise sheets and exams

Here is an example :

```
\begin{questions} % Begins the questions environment
\question[2] What is the solution? % Introduces a new question which is worth 2
 points
\begin{solution}
Here is the solution
\end{solution}
\question[5] What is your opinion?
\begin{solution}
This is my opinion
\end{solution}
\end{questions}
```

It is also possible to add stuff only if answers are printed using the **\ifprintanswers** command.

```
\ifprintanswers
Only if answers are printed
\else
Only if answers are not printed
\fi
```

42.2.3. Introduction

The macro **\numquestions** gives the total number of questions. The macro **\numpoints** gives the total number of points.

```
\begin{minipage}{.8\textwidth}
This exam includes \numquestions\ questions. The total number of points is
 \numpoints.
\end{minipage}
```

The backslash after **\numquestion** prevents the macro from gobbling the following whitespace as it normally would.

42.3. References

43. Curriculum Vitae

A *curriculum vitæ* or résumé has a universal requirement: its formatting must be flawless. This is a great example of cases where the power of LaTeX comes to the front. Thanks to its strong typographical stance, LaTeX is definitely a document processor of choice to write a CV.
Of course you can design your own CV by hand. Otherwise, you may want to use a dedicated class for that task. This way, writing a CV in LaTeX is as simple as filling the forms, and you are done. Seeveeze makes 3 of them available (ModernCV PlasmatiCV and FriggeriCV) from a simple web form: no coding or editor required.
A full list of CV packages is available at CTAN[1].

43.1. curve

43.2. europecv

```
\documentclass[utf8, a4paper, 10pt, helvetica, narrow, flagWB, booktabs,
 totpages, english]{europecv}
\usepackage{graphicx}                    % Required to draw the flag
\usepackage[a4paper, left=3cm, right=2cm, top=2cm, bottom=2cm]{geometry}
\usepackage{babel}

% Commands europecv

\ecvLogoWidth{12mm}                          % Size logo europass
%\ecvLeftColumnWidth{4cm}                  % Size of column and vertical line
 (different from standard)
%\ecvfootnote{footnote}                  % Foot notes
\ecvname{\textsc{Surname}, First Name}

% Personal picture

\ecvbeforepicture{\raggedleft}
\ecvpicture[height=1in]{namefile_pic}   % File picture without extension

\ecvafterpicture{\ecvspace{-2.5cm} }

% Address

\ecvaddress{Address first line\\& Address second line\\& City, State}

% Telephone

\ecvtelephone{+44 (0) 123 4567}
%\ecvfax{+39 01234567}

\ecvemail{john@someserver}

% Other personal info
```

1 http://www.ctan.org/topic/cv

```
\ecvnationality{Nationality}
\ecvdateofbirth{01/01/1900}
\ecvgender{Male}

\begin{document}

% Begin europecv environment

\begin{europecv}

\ecvpersonalinfo               % Print personal info in preamble

\ecvitem{}{}               % 1 free line - \ecvitem{}{} adds elements to a section
%\ecvsection{}               % \ecvsection{} adds sections

\ecvitem{\large\textbf{Desired employment / Occupational field}
 }{\Large\textbf{Dream job} }  % desired job

% Sections

% School

\ecvsection{Education and training}

\ecvitem{Dates}{From September 1900 to August 1905}\\
\ecvitem{Title of qualification awarded}{Name of the\\& degree}\\
\ecvitem{Principal subjects/occupational skills covered}{Learned skills}\\
\ecvitem{Name and type of organisation providing education and training}{My
 University\\&
Address\\&
City\\& Nation\\&
Post code\\&
Tel. +44 (0) 123 45678 23}\\
\ecvitem{Level in national or international classification}{Level of degree}\\

%\pagebreak{}

% Single course

\ecvitem{Dates}{August 2013}
\ecvitem{Title of qualification awarded}{Name of certification}
\ecvitem{Principal subjects/occupational skills covered}{Skills of
 certification}
\ecvitem{Name and type of organisation providing education and
 training}{Institution}\\

% Last working experience

\ecvsection{Work Experience}
\ecvitem{Dates}{From June 1957 to February 1987}\\
\ecvitem{Occupation or position held}{Name of the job}\\
\ecvitem{Main activities and responsibilities}{Activities during \\& this job}
\ecvitem{Name and address of employer}{Name of employer\\&
Employer address\\&
Second line\\& City\\& Nation\\&
Tel. +39 (0) 1234 5678}\\
\ecvitem{Type of business or sector}{Business}\\

% Volunteer experiences

\ecvsection{Volunteer Experience}

\ecvitem{Dates}{From August 2000 to present}\\
\ecvitem{Occupation or position held}{First Aider}\\
\ecvitem{Main activities and responsibilities}{Activities}
\ecvitem{Name and address of employer}{Name\\&
Address\\&
```

```
City\\& Post code\\&
Nation\\&
Tel. +44 (0) 1234 7654}\\
\ecvitem{Type of business or sector}{Business}\\

% Personal competences

\ecvsection{Personal skills and competences}

% Lenguages

% Mothertongue

\ecvmothertongue[10pt]{Italian}\\               % 10pt leave a one-char line space
 before the text

% Table for common lenguage evaluation

\ecvlanguageheader{(*)}
\ecvlanguage{English}{\ecvCOne}{\ecvCOne}{\ecvCOne}{\ecvCOne}{\ecvCOne}
    % second language and levels
      % Language levels A1 - A2 - B1 - B2 - C1 - C2 from basic to advanced.
      % in this package are \ecv + A, B or C and the sub-level in letters (One
 or Two)
\ecvlanguage{French}{\ecvBTwo}{\ecvBTwo}{\ecvBTwo}{\ecvBTwo}{\ecvBTwo}       %
 third
\ecvlastlanguage{Russian}{\ecvAOne}{\ecvATwo}{\ecvBOne}{\ecvCTwo}{\ecvBTwo}
% last language

\ecvlanguagefooter{(*)}\\

% Social skills

\ecvitem{Social skills and competences}{- First social skill;\\& - Second social
 skill}\\

% Technical skills

\ecvitem{Technical skills and competences}{- First technical skill;\\& - Second
 technical skill}\\

% Computer skills

\ecvitem{Computer skills and competences}{- First skill;\\& - Second}\\

% Other skills

\ecvitem{Other skills and competences}{- First otherskill}\\

% Driving Licence

\ecvitem{Driving licence(s)}{Category and Type}\\

% Annexes

\ecvsection{Annexes}
\ecvitem{}{On request:}
\ecvitem{}{Birth certificate}
\ecvitem{}{Passport}
\ecvitem{}{Driving licence}
\ecvitem{}{Criminal record certificate}
\ecvitem{}{University study plan}
\ecvitem{}{}

% Disclaimer

\ecvsection{Disclaimer}
\ecvitem{}{This informations may be used for all purposes permitted by law and
```

```
 under the Data Protection Act 1998.\\&
Autorizzo l'utilizzo dei dati personali contenuti nel presente curriculum ai
 sensi del D.Lgs. 196/2003 e s.m.i. (Codice in materia di protezione dei dati
 personali)}

\end{europecv}
\end{document}
```

43.3. moderncv

From CTAN:
Moderncv provides a documentclass for typesetting modern curriculums vitae, both in a classic and in a casual style. It is fairly customizable, allowing you to define your own style by changing the colours, the fonts, etc.
The official package provides some well commented templates which may be a good start. You can find those templates in your distribution (if documentation is installed along packages) or ultimately on CTAN[2].
We will not repeat the templates here, so we will only provide a crash course. You should really have a look at the templates for more details.

43.3.1. First document

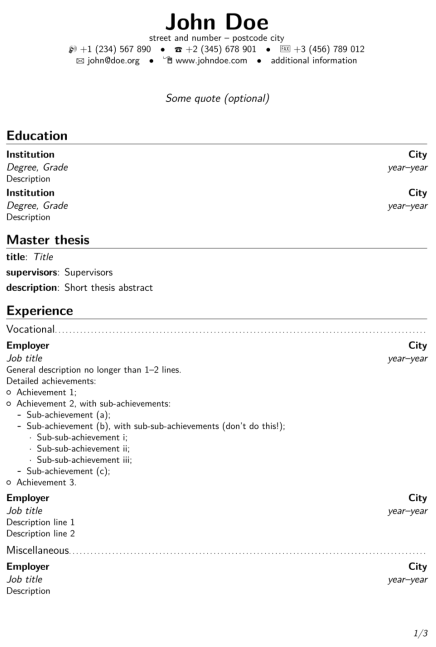
Most commands are self-explanatory.

```
\documentclass[11pt,a4paper,sans]{moderncv}

%% ModernCV themes
\moderncvstyle{casual}
\moderncvcolor{blue}
\renewcommand{\familydefault}{\sfdefault}
\nopagenumbers{}

%% Character encoding
\usepackage[utf8]{inputenc}

%% Adjust the page margins
\usepackage[scale=0.75]{geometry}

%% Personal data
\firstname{John}
\familyname{Doe}
\title{Resumé title (optional)}
\address{street and number}{postcode city}
\mobile{+1~(234)~567~890}
\phone{+2~(345)~678~901}
\fax{+3~(456)~789~012}
\email{john@doe.org}
\homepage{www.johndoe.com}
\extrainfo{additional information}
\photo[64pt][0.4pt]{picture}
\quote{Some quote (optional)}

%%-----------------------------------------------------------------------------
%% Content
%%-----------------------------------------------------------------------------
\begin{document}
\makecvtitle
```

2 http://www.ctan.org/tex-archive/macros/latex/contrib/moderncv/examples

```
\section{Education}
\cventry{year--year}{Degree}{Institution}{City}{ \textit{Grade} }{Description}
 % arguments 3 to 6 can be left empty
\cvitem{title}{ \emph{Title} }
\cvitemwithcomment{Language 1}{Skill level}{Comment}
\cvdoubleitem{category X}{XXX, YYY, ZZZ}{category Y}{XXX, YYY, ZZZ}
\cvlistitem{Item 1}
\cvlistdoubleitem{Item 2}{Item 3}
%% ...

\bibliography{publications}
\end{document}
```

43.3.2. Theme previews



John Doe
street and number – postcode city
+1 (234) 567 890 • +2 (345) 678 901 • +3 (456) 789 012
john@doe.org • www.johndoe.com • additional information

Some quote (optional)

Education

Institution **City**
Degree, Grade *year–year*
Description

Institution **City**
Degree, Grade *year–year*
Description

Master thesis

title: *Title*
supervisors: Supervisors
description: Short thesis abstract

Experience

Vocational..

Employer **City**
Job title *year–year*
General description no longer than 1–2 lines.
Detailed achievements:
- Achievement 1;
- Achievement 2, with sub-achievements:
  - Sub-achievement (a);
  - Sub-achievement (b), with sub-sub-achievements (don't do this!);
    - Sub-sub-achievement i;
    - Sub-sub-achievement ii;
    - Sub-sub-achievement iii;
  - Sub-achievement (c);
- Achievement 3.

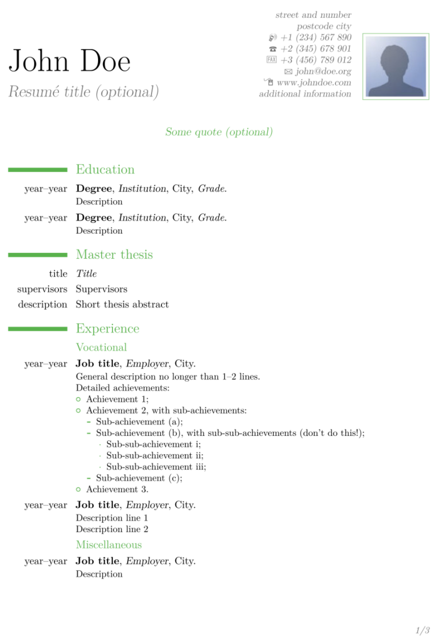
Employer **City**
Job title *year–year*
Description line 1
Description line 2

Miscellaneous..

Employer **City**
Job title *year–year*
Description

1/3

Figure 160 Banking black theme



street and number
postcode city
+1 (234) 567 890
+2 (345) 678 901
+3 (456) 789 012
john@doe.org
www.johndoe.com
additional information

John Doe

Resumé title (optional)

Some quote (optional)

Education

year–year **Degree**, *Institution*, City, *Grade*.
Description

year–year **Degree**, *Institution*, City, *Grade*.
Description

Master thesis

title *Title*
supervisors Supervisors
description Short thesis abstract

Experience

Vocational

year–year **Job title**, *Employer*, City.
General description no longer than 1–2 lines.
Detailed achievements:
- Achievement 1;
- Achievement 2, with sub-achievements:
  - Sub-achievement (a);
  - Sub-achievement (b), with sub-sub-achievements (don't do this!);
    - Sub-sub-achievement i;
    - Sub-sub-achievement ii;
    - Sub-sub-achievement iii;
  - Sub-achievement (c);
- Achievement 3.

year–year **Job title**, *Employer*, City.
Description line 1
Description line 2

Miscellaneous

year–year **Job title**, *Employer*, City.
Description

1/3

Figure 161 Classic green theme

43.4. Multilingual support

It is especially convenient for résumés to have only one document for several output languages, since many parts are shared among versions (personal data, structure, etc.). LaTeX with appropriate macros provide a comfortable way to manage it. See Internationalization[3].

43.5. References

3 Chapter 12 on page 133

Part VII.

Creating Graphics

44. Introducing Procedural Graphics

In the Importing Graphics[1] chapter, you learned that you can import or link graphics into LaTeX, such as graphics that you have created in another program or obtained elsewhere. In this chapter, you will learn how to create or embed graphics directly in a LaTeX document. The graphics is marked up using commands similar to those for typesetting bold text or creating mathematical formulas, as the following example of embedded graphics shows:

```
\begin{displaymath}
\xymatrix{ \bullet \ar[r] \ar@{.>}[r] & \bullet }
\end{displaymath}
```

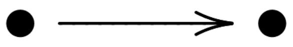


Figure 162

There are several packages supporting the creation of graphics directly in LaTeX, including `picture` [2], `xy-Pic` [3] and `PGF/TikZ` [4], described in the following sections.
Compared to WYSIWIG tools like Xfig or Inkscape, this approach is more time consuming, but leads to much better results. Furthermore, the ouput is flawlessly integrated to your document (no contrast in size nor fonts).
See the Importing Graphics[5] for more details on graphics importation and some attempts to circumvent to integration issue.

44.1. Overview

The `picture` environment allows programming pictures directly in LaTeX. On the one hand, there are rather severe constraints, as the slopes of line segments as well as the radii of circles are restricted to a narrow choice of values. On the other hand, the picture environment of LaTeX2e brings with it the `\qbezier` command, "q" meaning *quadratic* . Many frequently-used curves such as circles, ellipses, and catenaries[6] can be satisfactorily

1 Chapter 17 on page 211
2 Chapter 46 on page 521
3 Chapter 49 on page 559
4 Chapter 47 on page 535
5 Chapter 17 on page 211
6 `http://en.wikipedia.org/wiki/catenary`

approximated by quadratic Bézier curves, although this may require some mathematical toil. If a programming language like Java is used to generate `\qbezier` blocks of LaTeX input files, the picture environment becomes quite powerful.
Although programming pictures directly in LaTeX is severely restricted, and often rather tiresome, there are still reasons for doing so. The documents thus produced are "small" with respect to bytes, and there are no additional graphics files to be dragged along.
Packages like `epic` , `eepic` or `pstricks` enhance the original picture environment, and greatly strengthen the graphical power of LaTeX.
While the former two packages just enhance the picture environment, the `pstricks` package has its own drawing environment, `pspicture` . The power of `pstricks` stems from the fact that this package makes extensive use of PostScript possibilities. Unfortunately it has one big shortcoming: it doesn't work together with pdfLaTeX, as such. To generate a PDF document from TeX source, you have to go from TeX to DVI to PDF, losing hyperlinks, metadata, and microtypographic features of pdflatex in the process.
In addition, numerous packages have been written for specific purposes. One of them is *XY-pic,* described at the end of this chapter. A wide variety of these packages are described in detail in *The LaTeX Graphics Companion* (not to be confused with *The LaTeX Companion*).
Perhaps the most powerful graphical tool related with LaTeX is MetaPost[7], the twin of Donald E. Knuth's METAFONT[8]. MetaPost has the very powerful and mathematically sophisticated programming language of METAFONT. Contrary to METAFONT, which generates bitmaps, MetaPost generates encapsulated PostScript files, which can be imported in LaTeX. For an introduction, see *A User's Manual for MetaPost.* A very thorough discussion of LaTeX and TEX strategies for graphics (and fonts) can be found in *TEX Unbound.*
The last but certainly not least are the PGF/TikZ and Asymptote systems. While the previous systems (`picture` , `epic` , `pstricks` or `metapost`) focus on the *how* to draw, TikZ and Asymptote focus more on the *what* to draw. One could say that TikZ and Asymptote are to drawing in LaTeX as LaTeX is to digital typesetting. It's recommended to use one of these if your LaTeX distribution includes it. TikZ is a pure (La)TeX system, not reliant on external software, while Asymptote[9] is an external system which integrates seamlessly with (La)TeX. If using Asymptote, it is very helpful to use latexmk[10] to manage the compilation steps.
In many cases, especially for more advanced diagrams, it may be easier to draw the graphics using external vector graphics software, and then import the file into the document (see ../Importing Graphics[11]). However most software does not support LaTeX fonts or mathematical notation, which can result in not suitable and inconsistent graphics. There are several solutions to this problem.

7 `http://en.wikipedia.org/wiki/MetaPost`
8 `http://en.wikipedia.org/wiki/METAFONT`
9 `http://en.wikipedia.org/wiki/Asymptote%20%28vector%20graphics%20language%29`
10 `http://www.ctan.org/pkg/latexmk/`
11 Chapter 17 on page 211

45. MetaPost

46. Picture

The `picture` environment allows programming pictures directly in LaTeX. On the one hand, there are rather severe constraints, as the slopes of line segments as well as the radii of circles are restricted to a narrow choice of values. On the other hand, the picture environment of LaTeX2e brings with it the `\qbezier` command, "q" meaning *quadratic* . Many frequently-used curves such as circles, ellipses, and catenaries[1] can be satisfactorily approximated by quadratic Bézier curves, although this may require some mathematical toil. If a programming language like Java is used to generate `\qbezier` blocks of LaTeX input files, the picture environment becomes quite powerful.
Although programming pictures directly in LaTeX is severely restricted, and often rather tiresome, there are still reasons for doing so. The documents thus produced are "small" with respect to bytes, and there are no additional graphics files to be dragged along.
Packages like `pict2e`, `epic`, `eepic` or `pstricks` enhance the original picture environment, and greatly strengthen the graphical power of LaTeX.

46.1. Basic commands

A `picture` environment is available in any LaTeX distribution, without the need of loading any external package. This environment is created with one of the two commands

```
\begin{picture}(x, y)
 ...
\end{picture}
```

or

```
\begin{picture}(x, y)(x0, y0)
...
\end{picture}
```

The first pair, (x,y), affects the reservation, within the document, of rectangular space for the picture.
The optional second pair, (x_0,y_0), assigns arbitrary coordinates to the bottom left corner of the reserved rectangle.
The numbers x , y , x 0, y 0 are numbers (lengths) in the units of `\unitlength`, which can be reset any time (but not within a picture environment) with a command such as

```
\setlength{\unitlength}{1.2cm}
```

The default value of `\unitlength` is `1pt`.
Most drawing commands have one of the two forms

```
\put(x, y){object}
```

or

```
\multiput(x, y)(dx, dy){n}{object}
```

Bézier curves are an exception. They are drawn with the command

1 http://en.wikipedia.org/wiki/catenary

```
\qbezier(x1, y1)(x2, y2)(x3, y3)
```

With the package `picture` absolute dimension (like 15pt) and expression are allowed, in addition to numbers relative to `\unitlength`.

46.2. Line segments

Line segments are drawn with the command:

```
\put(x, y){ \line(x1, y1){length} }
```

The `\line` command has two arguments:

1. a direction vector,
2. a "length" (sort of: this argument is the vertical length in the case of a vertical line segment and in all other cases the horizontal distance of the line, rather than the length of the segment itself).

The components of the direction vector are restricted to the integers (*−6, −5, ... , 5, 6*) and they have to be coprime (no common divisor except 1). The figure below illustrates all 25 possible slope values in the first quadrant. The length is relative to `\unitlength`.

```
\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
\put(0,0){\line(0,1){1}}
\put(0,0){\line(1,0){1}}
\put(0,0){\line(1,1){1}}
\put(0,0){\line(1,2){.5}}
\put(0,0){\line(1,3){.3333}}
\put(0,0){\line(1,4){.25}}
\put(0,0){\line(1,5){.2}}
\put(0,0){\line(1,6){.1667}}
\put(0,0){\line(2,1){1}}
\put(0,0){\line(2,3){.6667}}
\put(0,0){\line(2,5){.4}}
\put(0,0){\line(3,1){1}}
\put(0,0){\line(3,2){1}}
\put(0,0){\line(3,4){.75}}
\put(0,0){\line(3,5){.6}}
\put(0,0){\line(4,1){1}}
\put(0,0){\line(4,3){1}}
\put(0,0){\line(4,5){.8}}
\put(0,0){\line(5,1){1}}
\put(0,0){\line(5,2){1}}
\put(0,0){\line(5,3){1}}
\put(0,0){\line(5,4){1}}
\put(0,0){\line(5,6){.8333}}
\put(0,0){\line(6,1){1}}
\put(0,0){\line(6,5){1}}
\end{picture}
```

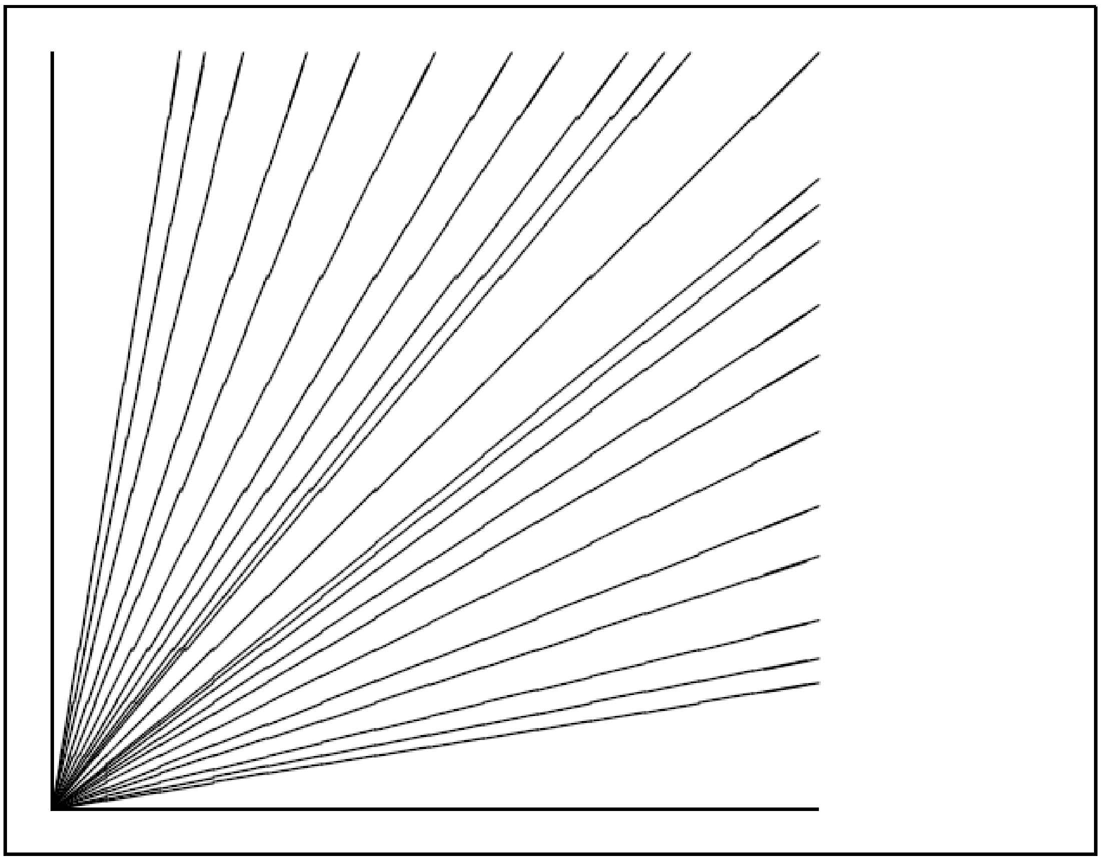


Figure 163

46.3. Arrows

Arrows are drawn with the command

```
\put(x, y){\vector(x1, y1){length}}
```

For arrows, the components of the direction vector are even more narrowly restricted than for line segments, namely to the integers (*−4, −3, ... , 3, 4*). Components also have to be coprime (no common divisor except 1). Notice the effect of the `\thicklines` command on the two arrows pointing to the upper left.

```
\setlength{\unitlength}{0.75mm}
\begin{picture}(60,40)
\put(30,20){\vector(1,0){30}}
\put(30,20){\vector(4,1){20}}
\put(30,20){\vector(3,1){25}}
\put(30,20){\vector(2,1){30}}
\put(30,20){\vector(1,2){10}}
\thicklines
\put(30,20){\vector(-4,1){30}}
\put(30,20){\vector(-1,4){5}}
\thinlines
\put(30,20){\vector(-1,-1){5}}
\put(30,20){\vector(-1,-4){5}}
\end{picture}
```

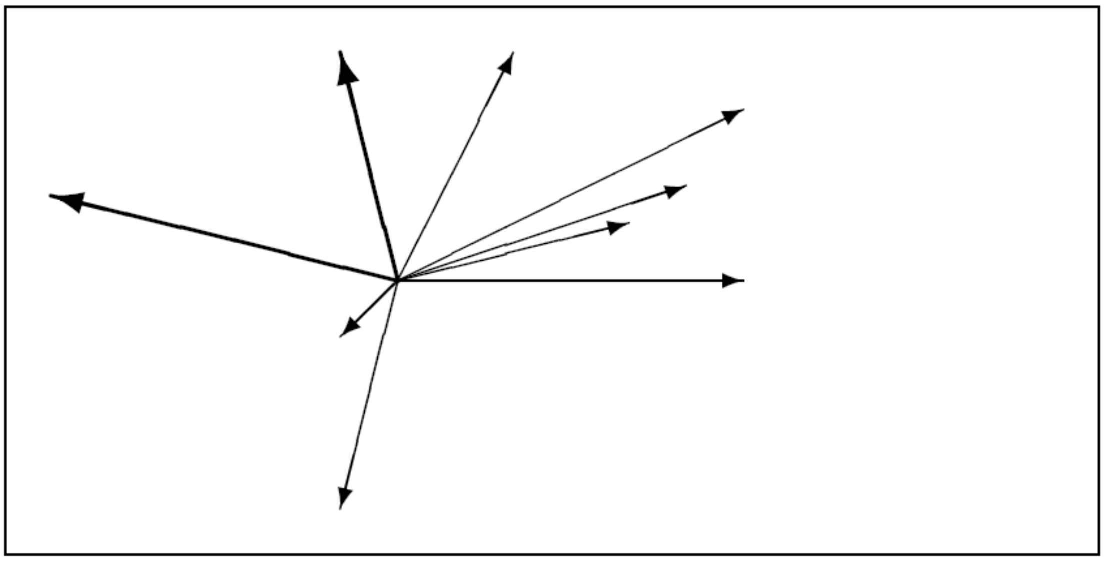


Figure 164

46.4. Circles

The command

```
\put(x, y){\circle{diameter}}
```

draws a circle with center (x, y) and diameter (not radius) specified by *diameter* . The picture environment only admits diameters up to approximately 14mm, and even below this limit, not all diameters are possible. The `\circle*` command produces disks (filled circles). As in the case of line segments, one may have to resort to additional packages, such as `eepic`, `pstricks`, or `tikz`.

```
\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
\put(20,30){\circle{1}}
\put(20,30){\circle{2}}
\put(20,30){\circle{4}}
\put(20,30){\circle{8}}
\put(20,30){\circle{16}}
\put(20,30){\circle{32}}
\put(40,30){\circle{1}}
\put(40,30){\circle{2}}
\put(40,30){\circle{3}}
\put(40,30){\circle{4}}
\put(40,30){\circle{5}}
\put(40,30){\circle{6}}
\put(40,30){\circle{7}}
\put(40,30){\circle{8}}
\put(40,30){\circle{9}}
\put(40,30){\circle{10}}
\put(40,30){\circle{11}}
\put(40,30){\circle{12}}
\put(40,30){\circle{13}}
\put(40,30){\circle{14}}
\put(15,10){\circle*{1}}
\put(20,10){\circle*{2}}
\put(25,10){\circle*{3}}
\put(30,10){\circle*{4}}
\put(35,10){\circle*{5}}
\end{picture}
```

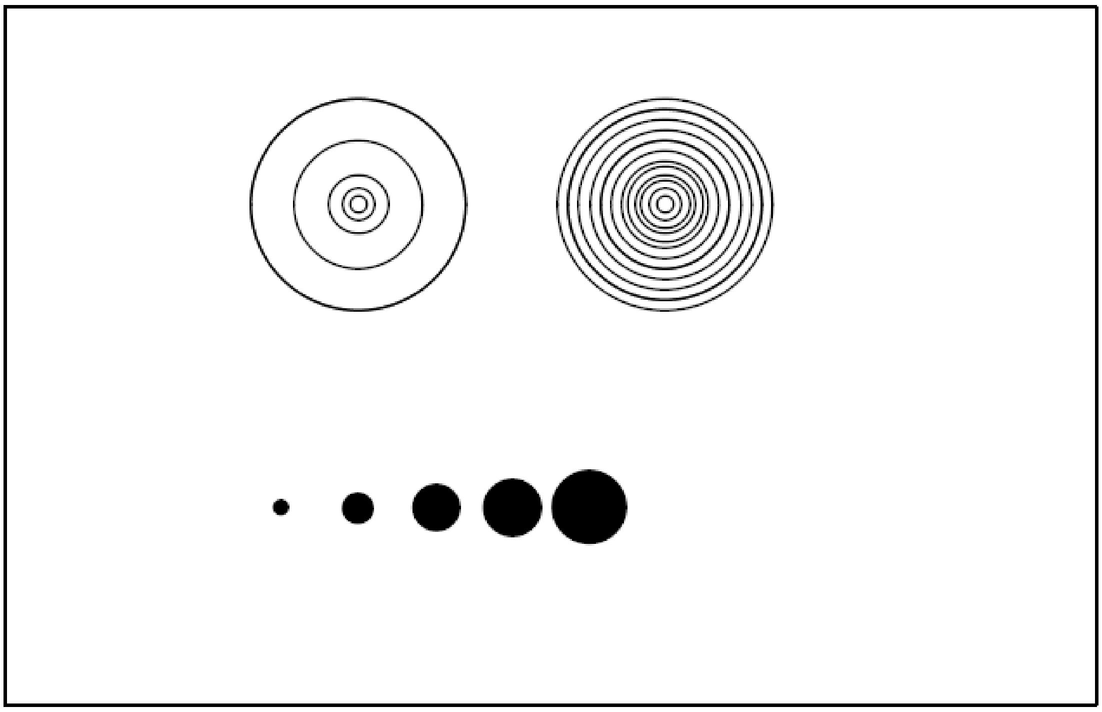


Figure 165

There is another possibility within the picture environment. If one is not afraid of doing the necessary calculations (or leaving them to a program), arbitrary circles and ellipses can be patched together from quadratic Bézier curves. See *Graphics in LaTeX2e* for examples and Java source files.

46.5. Text and formulae

As this example shows, text and formulae can be written in the environment with the `\put` command in the usual way:

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
\thicklines
\put(1,0.5){\line(2,1){3}}
\put(4,2){\line(-2,1){2}}
\put(2,3){\line(-2,-5){1}}
\put(0.7,0.3){$A$}
\put(4.05,1.9){$B$}
\put(1.7,2.95){$C$}
\put(3.1,2.5){$a$}
\put(1.3,1.7){$b$}
\put(2.5,1.05){$c$}
\put(0.3,4){$F=\sqrt{s(s-a)(s-b)(s-c)}$}
\put(3.5,0.4){$\displaystyle s:=\frac{a+b+c}{2}$}
\end{picture}
```

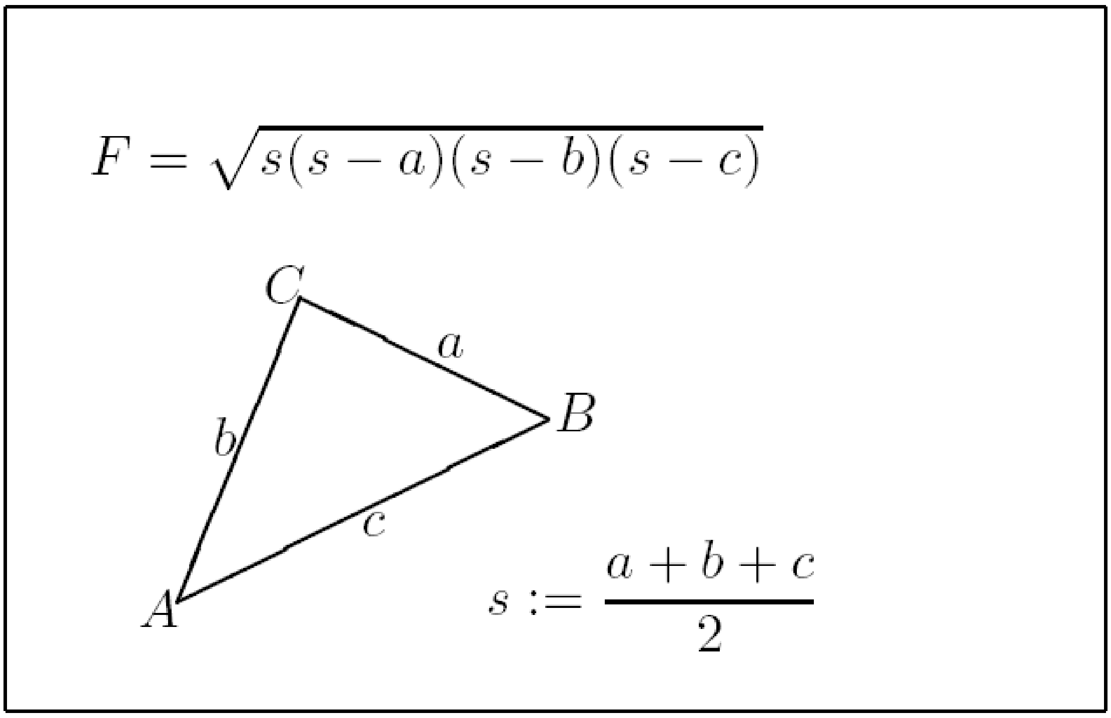


Figure 166

46.6. *\multiput* and *\linethickness*

The command

```
\multiput(x, y)(dx, dy ){n}{object}
```

has 4 arguments: the starting point, the translation vector from one object to the next, the number of objects, and the object to be drawn. The `\linethickness` command applies to horizontal and vertical line segments, but neither to oblique line segments, nor to circles. It does, however, apply to quadratic Bézier curves!

```
\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
\linethickness{0.075mm}
\multiput(0,0)(1,0){26}%
{\line(0,1){20}}
\multiput(0,0)(0,1){21}%
{\line(1,0){25}}
\linethickness{0.15mm}
\multiput(0,0)(5,0){6}%
{\line(0,1){20}}
\multiput(0,0)(0,5){5}%
{\line(1,0){25}}
\linethickness{0.3mm}
\multiput(5,0)(10,0){2}%
{\line(0,1){20}}
\multiput(0,5)(0,10){2}%
{\line(1,0){25}}
\end{picture}
```

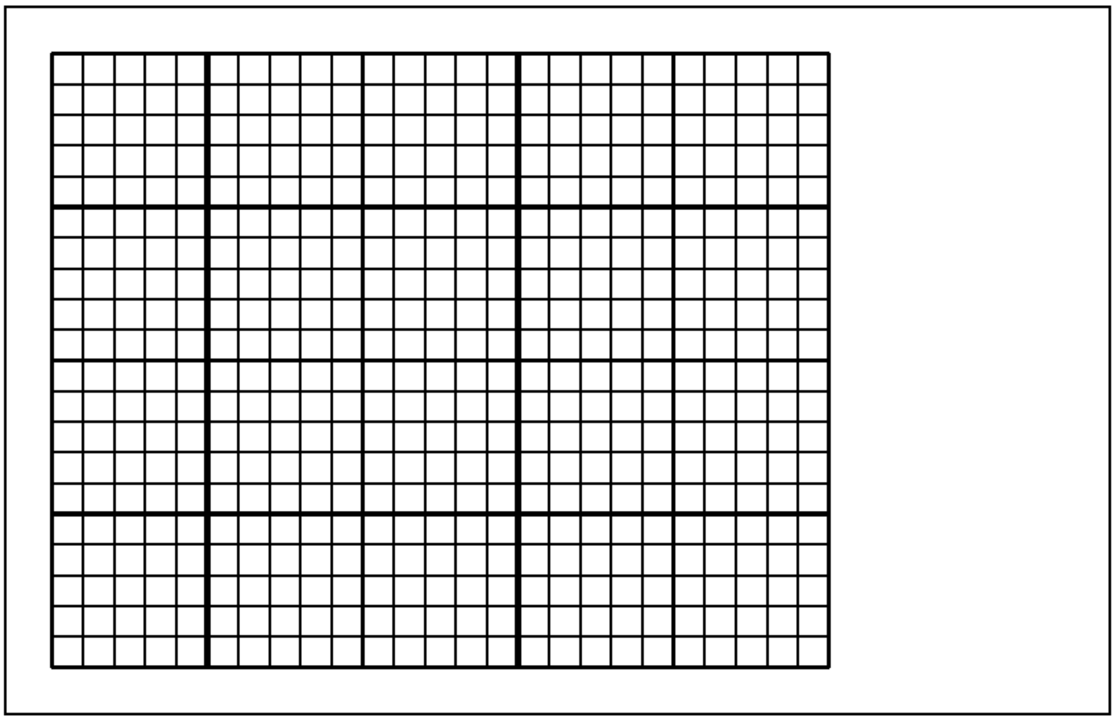


Figure 167

46.7. Ovals

The command

```
\put(x, y){\oval(w, h)}
```
or

```
\put(x, y){\oval(w, h)[position]}
```
produces an oval centered at *(x, y)* and having width *w* and height *h* . The optional position arguments *b, t, l, r* refer to ”top”, ”bottom”, ”left”, ”right”, and can be combined, as the example illustrates. Line thickness can be controlled by two kinds of commands: `\linethickness{''length''}` on the one hand, `\thinlines` and `\thicklines` on the other. While `\linethickness{''length''}` applies only to horizontal and vertical lines (and quadratic Bézier curves), `\thinlines` and `\thicklines` apply to oblique line segments as well as to circles and ovals.

```
\setlength{\unitlength}{0.75cm}
\begin{picture}(6,4)
\linethickness{0.075mm}
\multiput(0,0)(1,0){7}%
{\line(0,1){4}}
\multiput(0,0)(0,1){5}%
{\line(1,0){6}}
\thicklines
\put(2,3){\oval(3,1.8)}
\thinlines
\put(3,2){\oval(3,1.8)}
\thicklines
\put(2,1){\oval(3,1.8)[tl]}
\put(4,1){\oval(3,1.8)[b]}
\put(4,3){\oval(3,1.8)[r]}
\put(3,1.5){\oval(1.8,0.4)}
\end{picture}
```

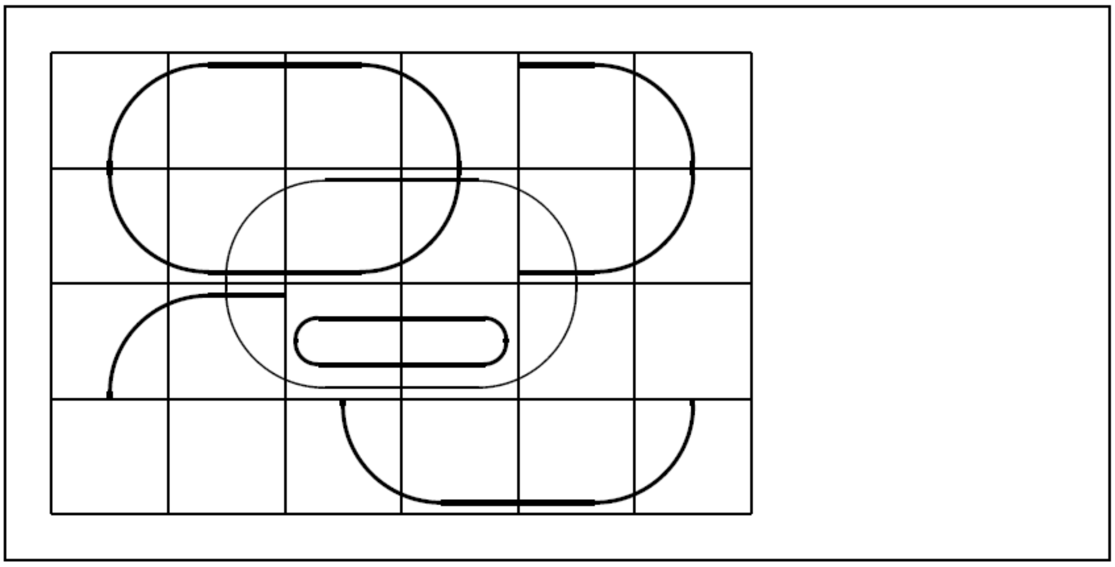


Figure 168

46.8. Multiple use of predefined picture boxes

A picture box can be *declared* by the command

```
\newsavebox{name}
```

then *defined* by

```
\savebox{name}(width,height)[position]{content}
```

and finally arbitrarily often be *drawn* by

```
\put(x, y){\usebox{name}}
```

The optional position parameter has the effect of defining the ”anchor point” of the savebox. In the example it is set to ”bl” which puts the anchor point into the bottom left corner of the savebox. The other position specifiers are top and right.
The *name* argument refers to a LaTeX storage bin and therefore is of a command nature (which accounts for the backslashes in the current example). Boxed pictures can be nested: In this example, `\foldera` is used within the definition of `\folderb`. The `\oval` command had to be used as the `\line` command does not work if the segment length is less than about 3 mm.

```
\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\foldera}
\savebox{\foldera}
  (40,32)[bl]{% definition
  \multiput(0,0)(0,28){2}
    {\line(1,0){40}<!---->}
  \multiput(0,0)(40,0){2}
    {\line(0,1){28}<!---->}
  \put(1,28){\oval(2,2)[tl]}
  \put(1,29){\line(1,0){5}<!---->}
  \put(9,29){\oval(6,6)[tl]}
  \put(9,32){\line(1,0){8}<!---->}
  \put(17,29){\oval(6,6)[tr]}
  \put(20,29){\line(1,0){19}<!---->}
  \put(39,28){\oval(2,2)[tr]}
}

\newsavebox{\folderb}
\savebox{\folderb}
  (40,32)[l]{% definition
  \put(0,14){\line(1,0){8}<!---->}
  \put(8,0){\usebox{\foldera}<!---->}
}

\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
{\usebox{\folderb}}
\end{picture}
```

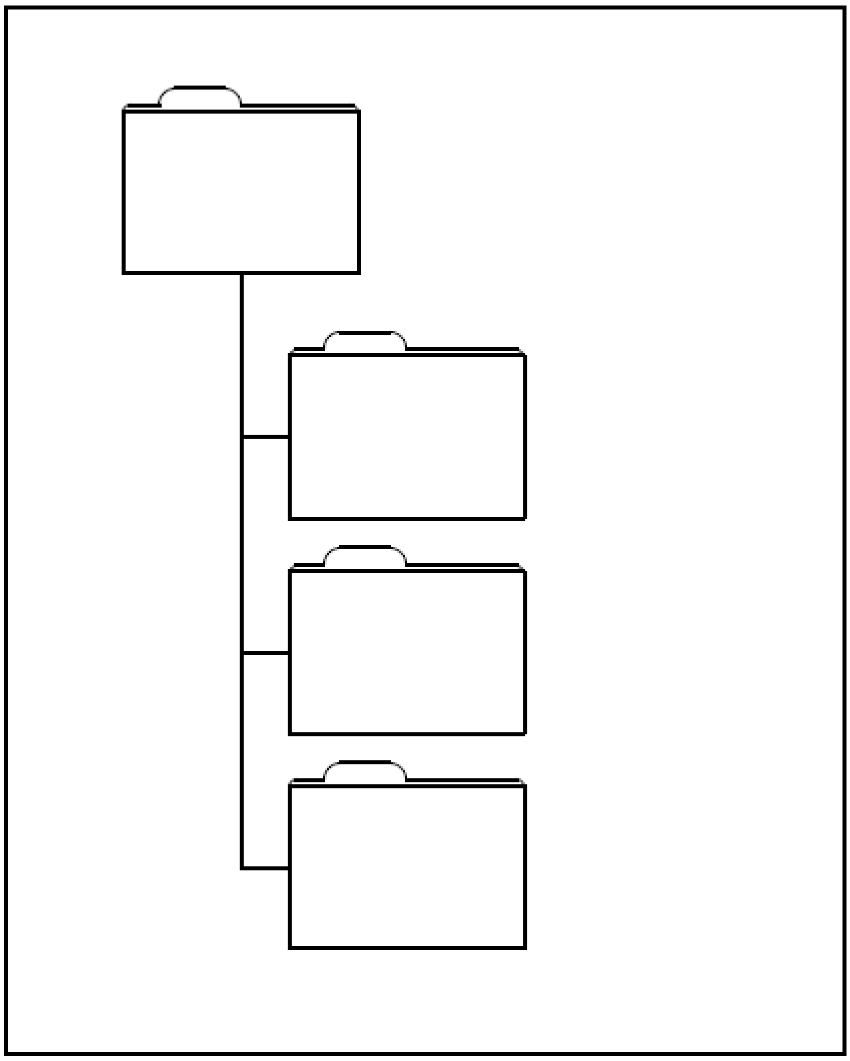


Figure 169

46.9. Quadratic Bézier curves

The command

```
\qbezier(x1, y1)(x, y)(x2, y2)
```

draws a quadratic bezier curve where $P_1 = (x_1,y_1)$, $P_2 = (x_2,y_2)$ denote the end points, and $S = (x,y)$ denotes the intermediate control point. The respective tangent slopes, m_1 and m_2, can be obtained from the equations

$$\begin{cases} x = \frac{m_2x_2-m_1x_1-(y_2-y_1)}{m_2-m_1} \\ y = y_i + m_i(x-x_i); \quad (i = 1,2 \text{ gives same solution}) \end{cases}$$

See *Graphics in LaTeX2e* for a Java program which generates the necessary `\qbezier` command line.

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)
\linethickness{0.075mm}
\multiput(0,0)(1,0){7}
{\line(0,1){4}}
\multiput(0,0)(0,1){5}
{\line(1,0){6}}
\thicklines
\put(0.5,0.5){\line(1,5){0.5}}
\put(1,3){\line(4,1){2}}
\qbezier(0.5,0.5)(1,3)(3,3.5)
\thinlines
\put(2.5,2){\line(2,-1){3}}
\put(5.5,0.5){\line(-1,5){0.5}}
\linethickness{1mm}
\qbezier(2.5,2)(5.5,0.5)(5,3)
\thinlines
\qbezier(4,2)(4,3)(3,3)
\qbezier(3,3)(2,3)(2,2)
\qbezier(2,2)(2,1)(3,1)
\qbezier(3,1)(4,1)(4,2)
\end{picture}
```

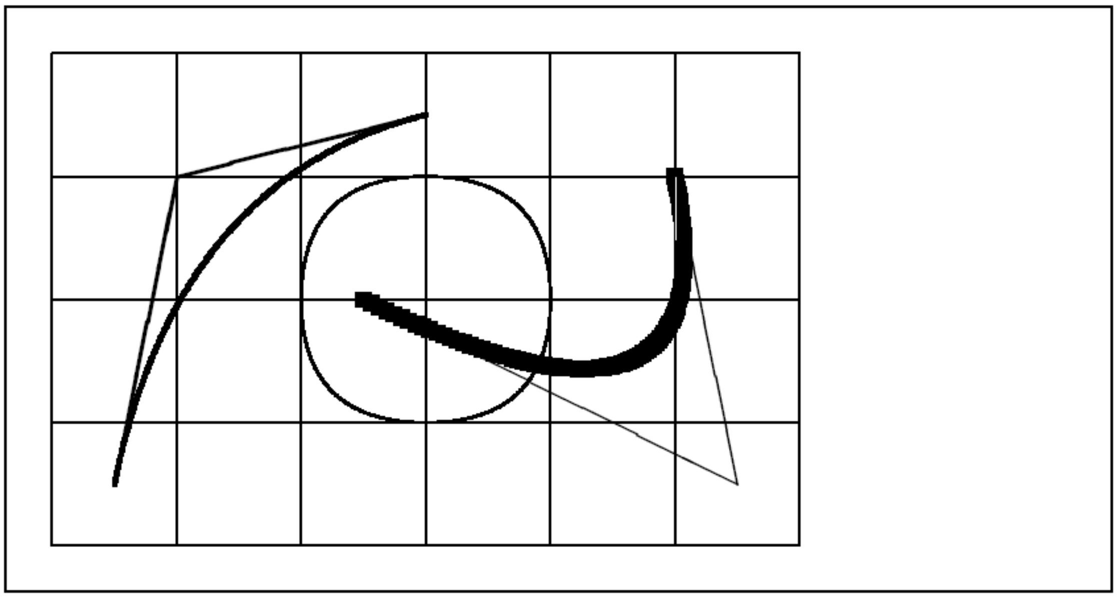


Figure 170

As this example illustrates, splitting up a circle into 4 quadratic Bézier curves is not satisfactory. At least 8 are needed. The figure again shows the effect of the `\linethickness` command on horizontal or vertical lines, and of the `\thinlines` and the `\thicklines` commands on oblique line segments. It also shows that both kinds of commands affect quadratic Bézier curves, each command overriding all previous ones.

46.10. Catenary

```
\setlength{\unitlength}{1cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
\put(-2,0){\vector(1,0){4.4}}
\put(2.45,-.05){$x$}
\put(0,0){\vector(0,1){3.2}}
\put(0,3.35){\makebox(0,0){$y$}}
\qbezier(0.0,0.0)(1.2384,0.0)
(2.0,2.7622)
\qbezier(0.0,0.0)(-1.2384,0.0)
(-2.0,2.7622)
\linethickness{.075mm}
\multiput(-2,0)(1,0){5}
{\line(0,1){3}}
\multiput(-2,0)(0,1){4}
{\line(1,0){4}}
\linethickness{.2mm}
\put( .3,.12763){\line(1,0){.4}}
\put(.5,-.07237){\line(0,1){.4}}
\put(-.7,.12763){\line(1,0){.4}}
\put(-.5,-.07237){\line(0,1){.4}}
\put(.8,.54308){\line(1,0){.4}}
\put(1,.34308){\line(0,1){.4}}
\put(-1.2,.54308){\line(1,0){.4}}
\put(-1,.34308){\line(0,1){.4}}
\put(1.3,1.35241){\line(1,0){.4}}
\put(1.5,1.15241){\line(0,1){.4}}
\put(-1.7,1.35241){\line(1,0){.4}}
\put(-1.5,1.15241){\line(0,1){.4}}
\put(-2.5,-0.25){\circle*{0.2}}
\end{picture}
```

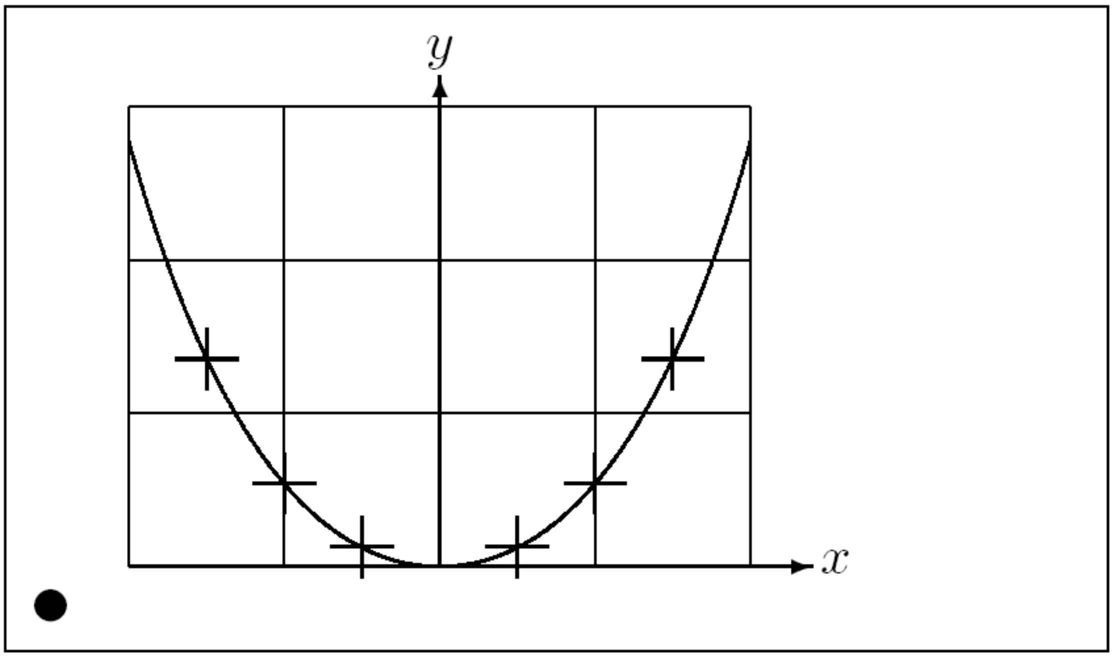


Figure 171

In this figure, each symmetric half of the catenary $y = \cosh x - 1$ is approximated by a quadratic Bézier curve. The right half of the curve ends in the point (2, 2.7622), the slope there having the value m = 3.6269. Using again equation (*), we can calculate the intermediate control points. They turn out to be (1.2384, 0) and (–1.2384, 0). The crosses indicate points of the real catenary. The error is barely noticeable, being less than one percent. This example points out the use of the optional argument of the `\begin{picture}` command. The picture is defined in convenient ”mathematical” coordinates, whereas by the command

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

its lower left corner (marked by the black disk) is assigned the coordinates (–2.5,–0.25).

46.11. Plotting graphs

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,6)(-3,-3)
\put(-1.5,0){\vector(1,0){3}}
\put(2.7,-0.1){$\chi$}
\put(0,-1.5){\vector(0,1){3}}
\multiput(-2.5,1)(0.4,0){13}
{\line(1,0){0.2}}
\multiput(-2.5,-1)(0.4,0){13}
{\line(1,0){0.2}}
\put(0.2,1.4)
{$\beta=v/c=\tanh\chi$}
\qbezier(0,0)(0.8853,0.8853)
(2,0.9640)
\qbezier(0,0)(-0.8853,-0.8853)
(-2,-0.9640)
\put(-3,-2){\circle*{0.2}}
\end{picture}
```

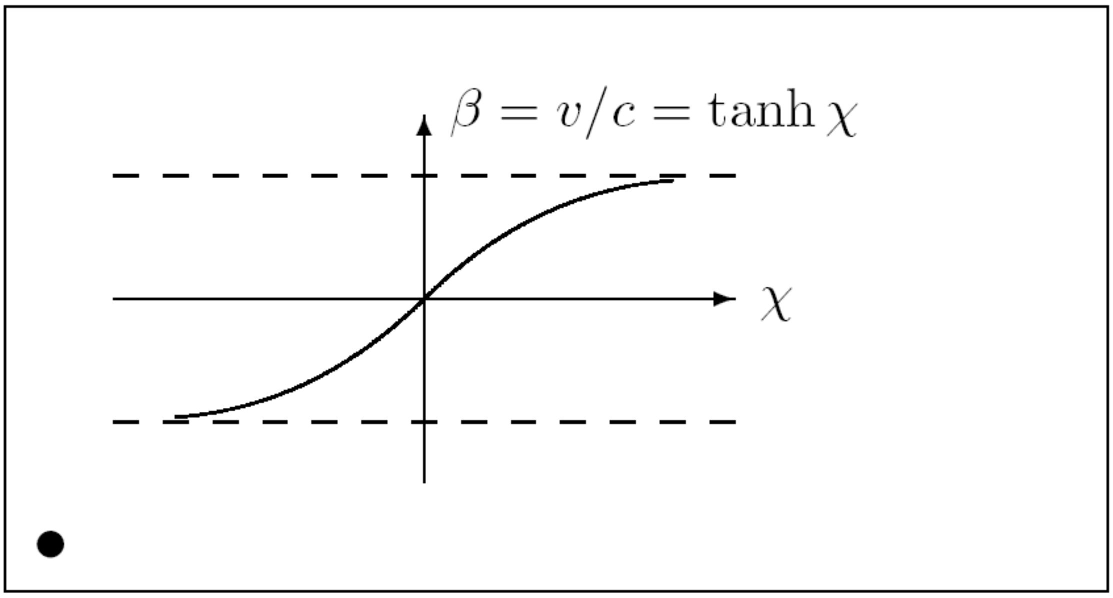


Figure 172

The control points of the two Bézier curves were calculated with formulas (*). The positive branch is determined by $P_1 = (0,0)$, $m_1 = 1$ and $P_2 = (2, \tanh 2)$, $m_2 = 1/\cosh^2 2$.

Again, the picture is defined in mathematically convenient coordinates, and the lower left corner is assigned the mathematical coordinates (–3,–2) (black disk).

46.12. The *picture* environment and gnuplot

The powerful scientific plotting package gnuplot[2] has the capability to output directly to a LaTeX `picture` environment. It is often far more convenient to plot directly to LaTeX, since this saves having to deal with potentially troublesome postscript files. Plotting scientific data (or, indeed, mathematical figures) this way gives much greater control, and of course typesetting ability, than is available from other means (such as postscript). Such pictures can then be added to a document by an `\include{}` command.
N.B. gnuplot is a powerful piece of software with a vast array of commands. A full discussion of gnuplot lies beyond the scope of this note. See `http://www.gnuplot.info/files/tutorial.pdf`[3]`http://` for a tutorial.

2 `http://en.wikipedia.org/wiki/gnuplot`
3 Chapter 1 on page 5

47. PGF/TikZ

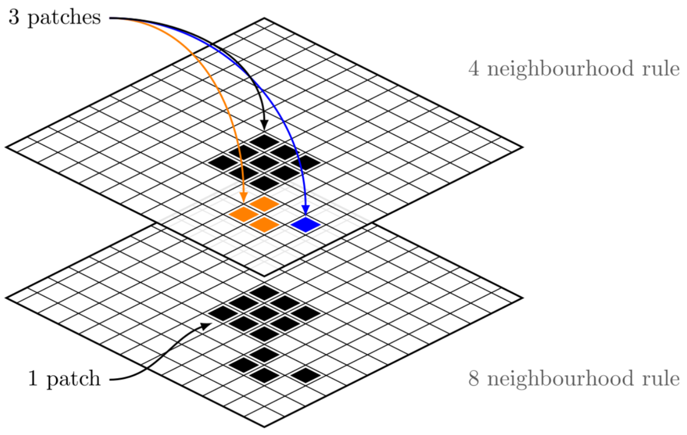


Figure 173 Example of graphics done with Ti*k* z.
Note the slightly translucent top layer.

One way to draw graphics directly with TeX commands is PGF/TikZ[1]. Ti*k* Z can produce portable graphics in both PDF and PostScript formats using either plain (pdf)TEX, (pdf)Latex or ConTEXt. It comes with very good documentation[2] and an extensive collection of examples: `http://www.texample.net/tikz/`
PGF (”portable graphics format”) is the basic layer, providing a set of basic commands for producing graphics, and Ti*k* Z (”Ti*k* Z ist *kein* Zeichenprogramm”) is the frontend layer with a special syntax, making the use of PGF easier. Ti*k* Z commands are prevalently similar to Metafont, the option mechanism is similar to PsTricks syntax.
While the previous systems (`picture` , `epic` , `pstricks` or `metapost`) focus on the *how* to draw, TikZ focuses more on the *what* to draw. One could say that TikZ is to drawing in LaTeX as LaTeX is to digital typesetting. It's recommended to use it if your LaTeX distribution includes it.
Other packages building on top of Ti*k* Z (e.g., for drawing electrical circuits) can be found here: `http://ftp.dante.de/tex-archive/help/Catalogue/bytopic.html#pgftikzsection`
In the following some basics of Ti*k* Z are presented.

47.1. Loading Package, Libraries - tikzpicture environment

Using Ti*k* Z in a LaTeX document requires loading the tikz package:

1 `http://en.wikipedia.org/wiki/PGF%2FTikZ`
2 `http://ftp.fau.de/ctan/graphics/pgf/base/doc/pgfmanual.pdf`

```
\usepackage{tikz}
```

somewhere in the preamble. This will automatically load the pgf package. To load further libraries use

```
\usetikzlibrary{⟨list of libraries separated by commas⟩}
```

Examples for libraries are ”`arrows` ”, ”`automata` ”, ”`backgrounds` ”, ”`calendar` ”, ”`chains` ”, ”`matrix` ”, ”`mindmap` ”, ”`patterns` ”, ”`petri` ”, ”`shadows` ”, ”`shapes.geometric` ”, ”`shapes.misc` ”, ”`spy` ”, ”`trees` ”.
Drawing commands have to be enclosed in an tikzpicture environment

```
\begin{tikzpicture}[⟨options⟩]
  ⟨tikz commands⟩
\end{tikzpicture}
```

or alternatively

```
\tikz[⟨options⟩]{⟨tikz commands⟩}
```

One possible option useful for inlined graphics is

```
baseline=⟨dimension⟩
```

Without that option the lower end of the picture is put on the baseline of the surrounding text. Using this option, you can specify that the picture should be raised or lowered such that the height ⟨dimension⟩is on the baseline.
Another option to scale the entire picture is

```
scale=⟨factor⟩
```

or different for height and width, e.g:

```
xscale=2.5, yscale=0.5
```

47.2. Specifying Coordinates

Coordinates are specified in round brackets in an arbitrary TEX dimension either using Cartesian coordinates (comma separated), e.g. 1cm in the x direction and 2pt in the y direction

```
(1cm,2pt)
```

or using polar coordinates (colon separated), e.g. 1cm in 30 degree direction

```
(30:1cm)
```

Without specifying a unit `(1,2)` , the standard one is cm `(1cm,2cm)` .
Relative coordinates to the previous given point are given by adding one or two plus signs in front of the coordinate. With ”++ ” the last point of the path becomes the current position, with ”+ ” the previous point stays the current path position. Example: 2 standard units to the right of the last point used:

```
++(2,0)
```

47.3. Syntax for Paths

A path is a series of straight and curved line segments(in a simplified explanation). The instruction has to end with a semicolon.

```
\path[<options>]⟨specification⟩;
```

One instruction can spread over several lines, or several instructions can be put on one line.
Options for path actions are e.g: ”`draw` ”, ”`fill` ”, ”`pattern` ”, ”`shade` ” (filling, in which its color changes smoothly from one to another), ”`clip` ” (all subsequent drawings up to the end of the current scope are clipped against the current path and the size of subsequent paths will not be important for the picture size), ”`use as bounding box` ”.
The ”`\path` ” command with these options can be combined to: ”`\draw` ”, ”`\fill` ”, ”`\filldraw` ”, ”`\pattern` ”, ”`\shade` ”, ”`\shadedraw` ”, ”`\clip` ”, ”`\useasboundingbox` ” .
Geometric path options: ”`rotate=<angle in degree>` ”, ”`xshift=<length>` ”, ”`yshift=<length>` ”, ”`scaling=<factor>` ”, ”`xscale=<factor>` ”, ”`yscale=<factor>` ”.
Color options for drawing paths: ”`color=<color name>` ”, ”`draw=<line color>` ”, ”`opacity=<factor>` ”. Following colors are predefined: red, green, blue, cyan , magenta, yellow, black, gray, darkgray, lightgray, brown, lime, olive, orange, pink, purple, teal, violet and white.
Line width options: ”`line width=<dimension>` ”, and abbreviations ”`ultra thin` ” for 0.1pt, ”`very thin` ” for 0.2pt, ”`thin` ” for 0.4pt (the default width), ”`semithick` ” for 0.6pt, ”`thick` ” for 0.8pt, ”`very thick` ” for 1.2pt, ”`ultra thick` ” for 1.6pt.
Line end, line join options: ”`line cap=<type: round, rect, or butt>` ”, ”`arrows=<start arrow kind>-<end arrow kind>` ”, ”`rounded corners` ”, ”`rounded corners=<size>` ”, ”`line join=<type: round, bevel, or miter>` ”.
Line pattern options: ”`dash pattern=<dash pattern>` ” (e.g. ”`dash pattern=on 2pt off 3pt on 4pt off 4pt` ”), ”`dash phase=⟨dash phase⟩` ”, ”`solid` ”, ”`dashed` ”, ”`dotted` ”, ”`dashdotted` ”, ”`densely dotted` ”, ”`loosely dotted` ”, ”`double` ”.
Options for filling paths are e.g. ”`fill=<fill color>` ”, ”`pattern=<name>` ”, ”`pattern color=<color>` ”
Straight lines are given by coordinates separated by a double minus,

```
\draw (1,0) -- (0,0) -- (0,1);
```

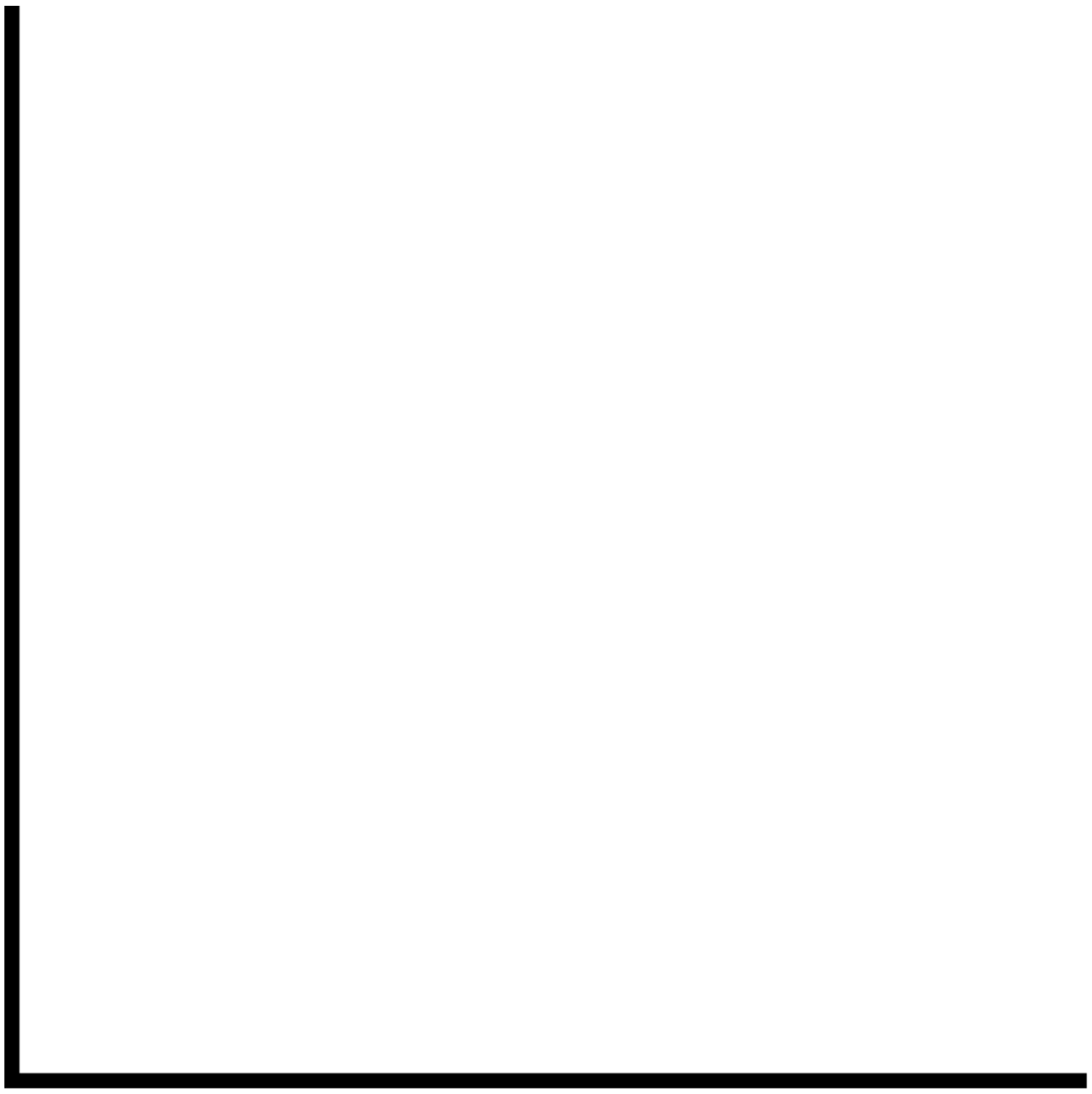


Figure 174

The first coordinate represents a move-to operation. This is followed by a series of “path extension operations”, like ”`-- (coordinates)` ”.
The same path with some drawing options:

```
\draw[red, dashed, very thick, rotate=30] (1,0) -- (0,0) -- (0,1);
```

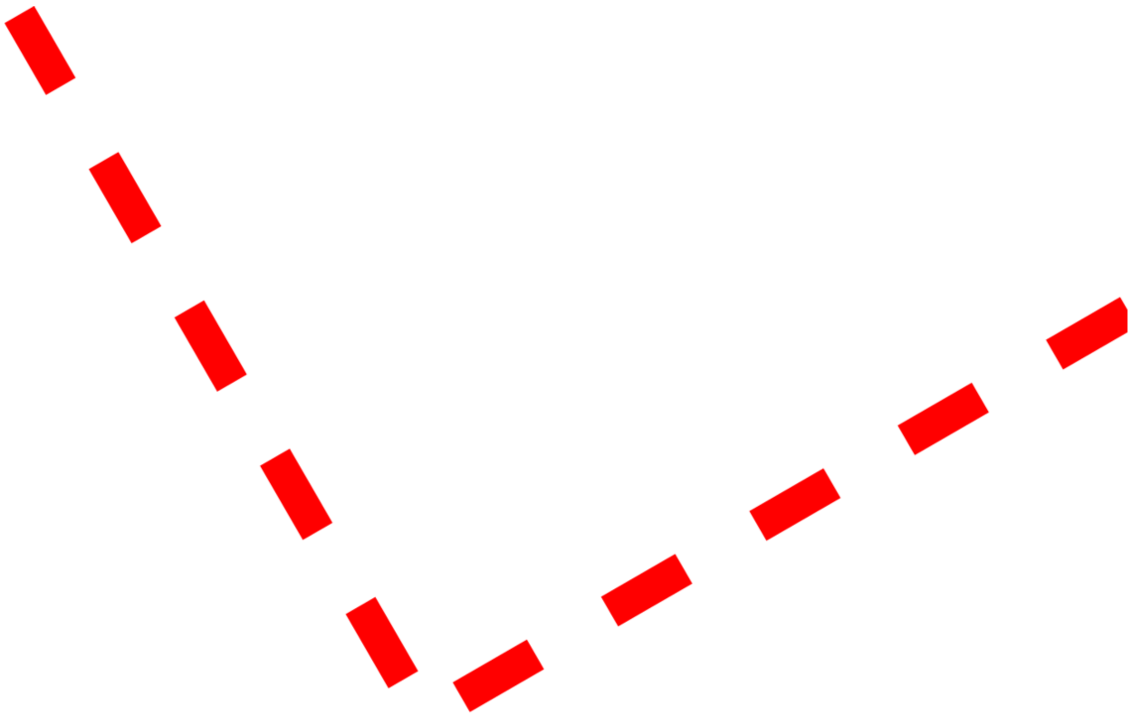


Figure 175

A connected path can be closed using the "`--cycle` " operation:

```
\draw (1,0) -- (0,0) -- (0,1) -- cycle;
```

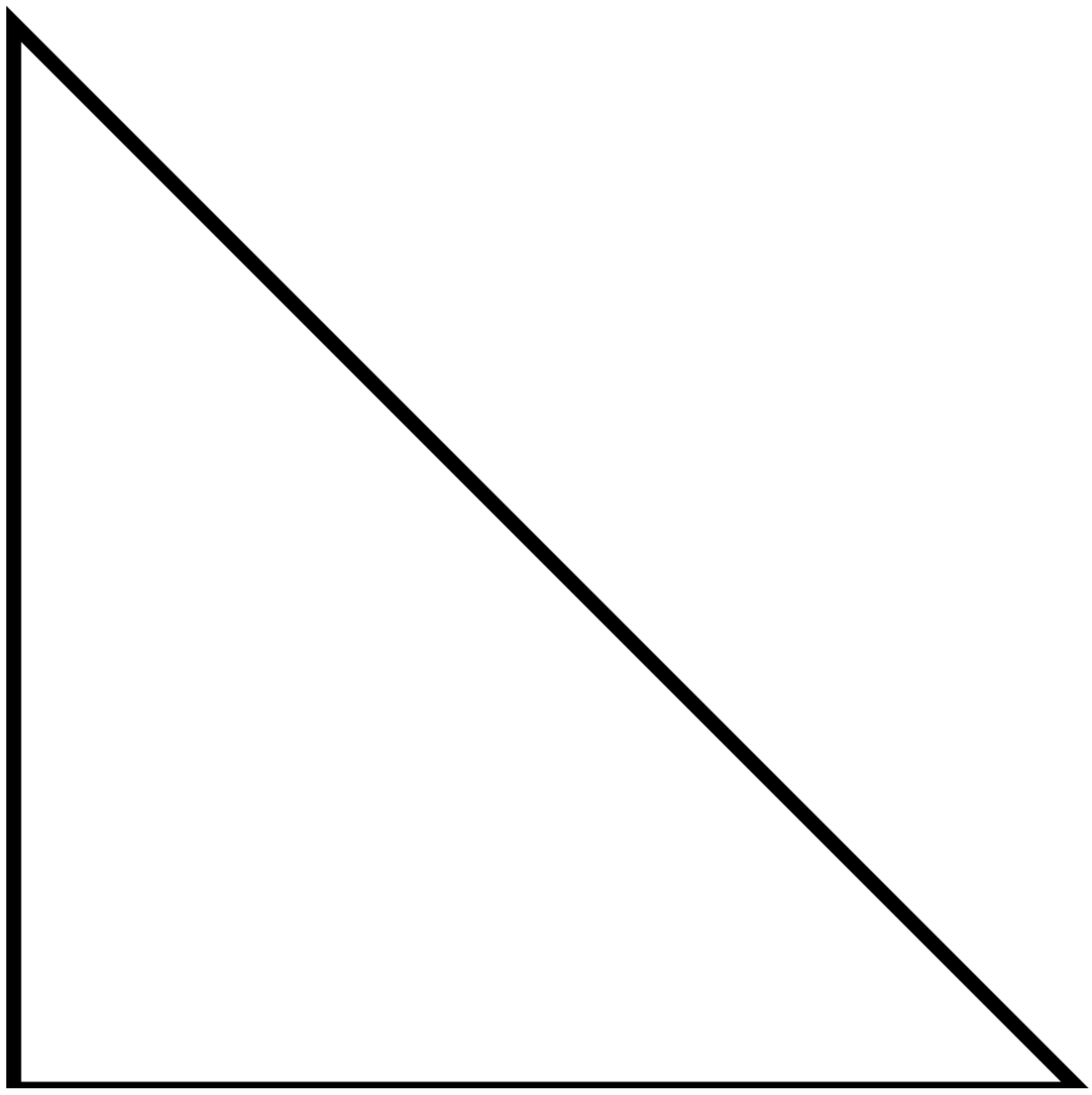


Figure 176

A further move-to operation in an existing path starts a new part of the path, which is not connected to the previous part of the path. Here: Move to (0,0) straight line to (2,0), move to (0,1) straight line to (2,1):

```
\draw (0,0) -- (2,0) (0,1) -- (2,1);
```

Figure 177

Connecting two points via straight lines that are only horizontal and vertical, use for first horizontal then vertical

```
\draw (0,0) -| (1,1);
```

or for first vertical then horizontal

```
\draw (0,0) |- (1,1);
```

Curved paths using a Bezier curve can be created using the ”`..controls() ..()` ” command, with one or two control points.

```
\draw (0,0) .. controls (1,1) .. (4,0)
      (5,0) .. controls (6,0) and (6,1) .. (5,2);
```

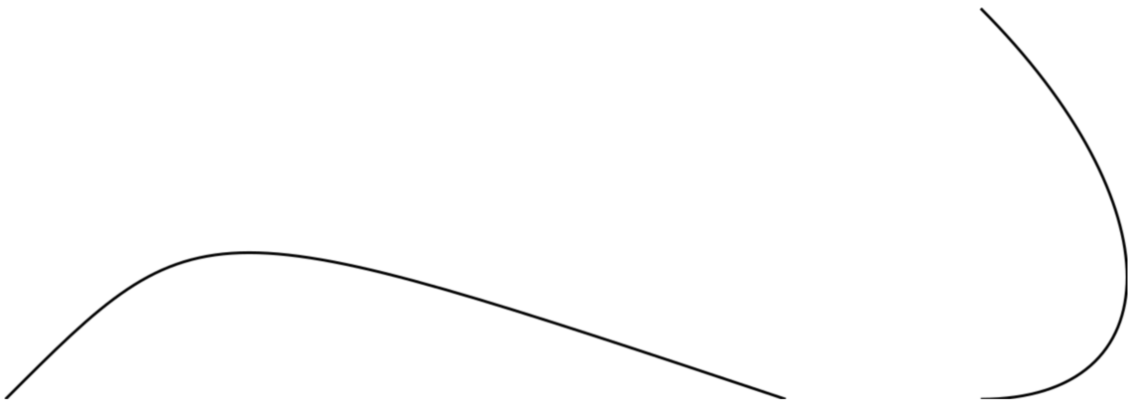


Figure 178

User-defined paths can be created using the ”`to` ” operation. Without an option it corresponds to a straight line, exactly like the double minus command. Using the ”`out` ” and ”`in` ” option a curved path can created. E.g. ”`[out=135,in=45]` ” causes the path to leave at an angle of 135 degree at the first coordinate and arrive at an angle of 45 degree at the second coordinate.

```
\draw (0,0) to (3,2);
\draw (0,0) to[out=90,in=180] (3,2);
\draw (0,0) to[bend right] (3,2);
```

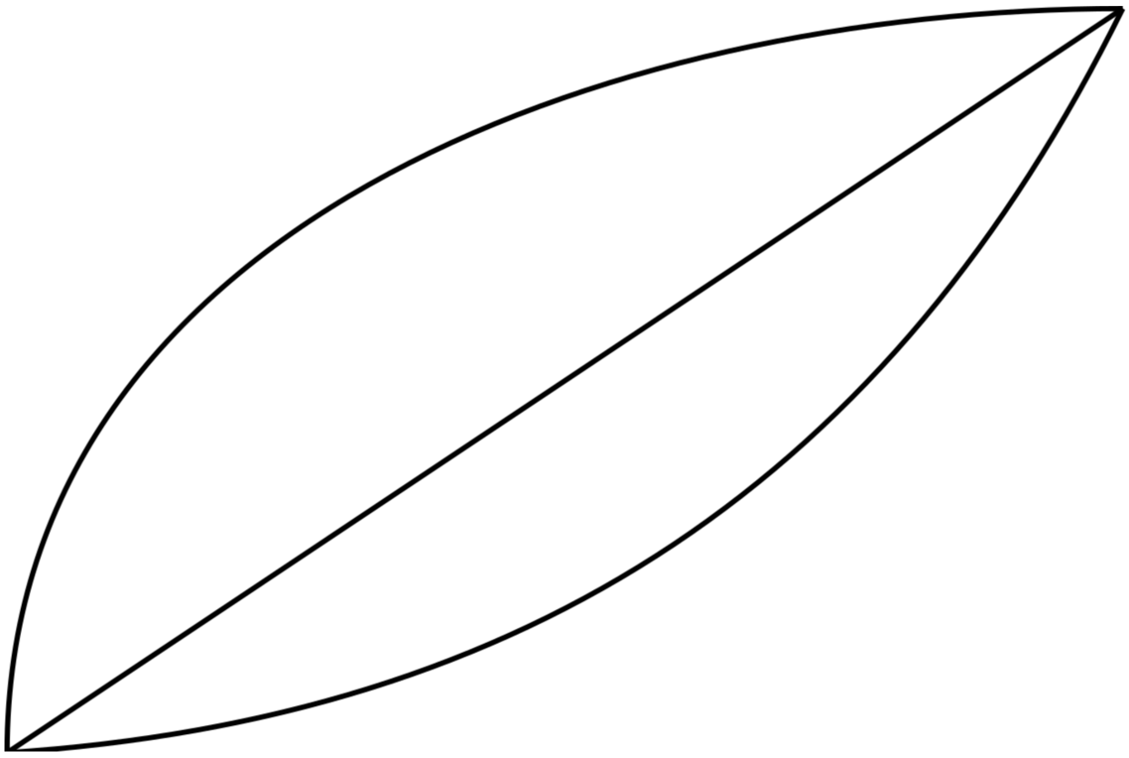


Figure 179

For rectangles a special syntax exist. Use a move-to operation to one corner and after "`rectangle` " the coordinates of the diagonal corner. The last one becomes the new current point.

```
\draw (0,0) rectangle (1,1);
\shade[top color=yellow, bottom color=black] (0,0) rectangle (2,-1);
\filldraw[fill=green!20!white, draw=green!40!black] (0,0) rectangle (2,1);
```

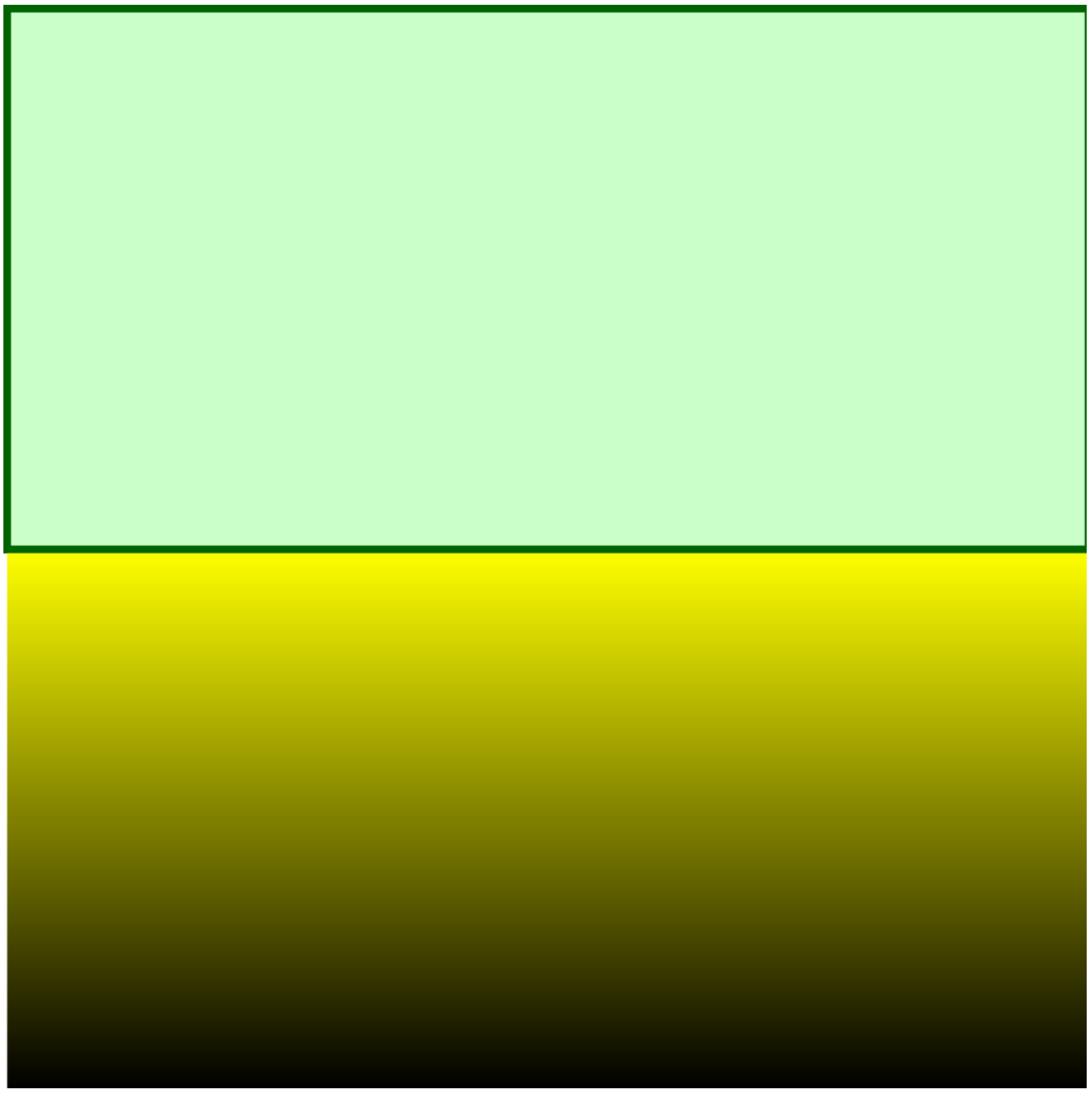


Figure 180

The fill color "`green!20!white` " means 20% green and 80% white mixed together. Circles and ellipses paths are defined beginning with their center then using the "`circle command` " either with one length as radius of a circle or with two lengths as semi-axes of an ellipse.

```
\draw (0,0) circle [radius=1.5];
\draw (0,0) circle (2cm); % old syntax
\draw (0,0) circle [x radius=1.5cm, y radius=10mm];
\draw (0,0) circle (1.2cm and 8mm); % old syntax
\draw (0,0) circle [x radius=1cm, y radius=5mm, rotate=30];
\draw[rotate=30] (0,0) ellipse (20pt and 10pt);  % old syntax
```

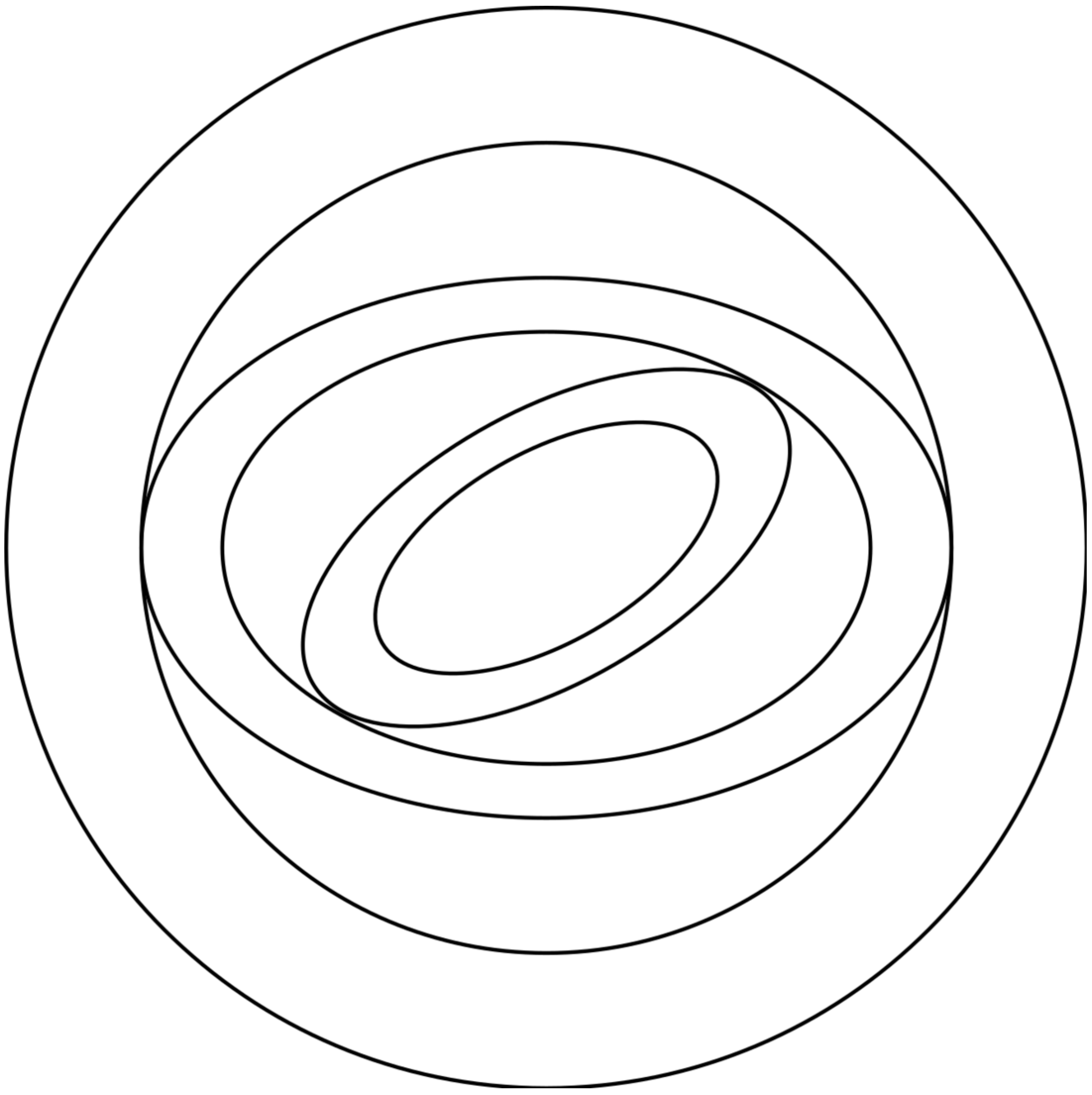


Figure 181

The command "`arc` " creates a part of a circle or an ellipse:

```
\draw (0,0) arc (0:270:8mm);
\draw (0,0) arc (0:315:1.75cm and 1cm);
\filldraw[fill=cyan, draw=blue] (0,0) -- (12mm,0mm) arc (0:30:12mm) -- (0,0);
```

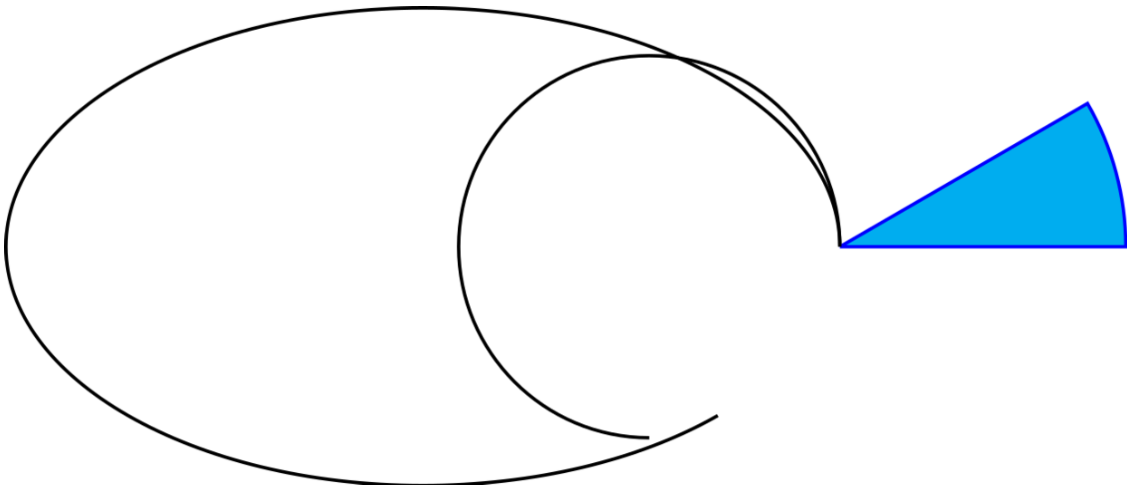


Figure 182

Or in an alternative syntax:

```
\draw (0,0)  arc[radius = 8mm, start angle= 0, end angle= 270];
\draw (0,0)  arc[x radius = 1.75cm, y radius = 1cm, start angle= 0, end angle=
 315];
```

There are many more predefined commands for special paths, like ”`grid` ”, ”`parabola` ”, ”`sin` ”, ”`cos` ” (sine or cosine curve in the interval [0,π/2]).

```
\draw[help lines] (0,0) grid (2,3);
\draw[step=0.5, gray, very thin] (-1.4,-1.4) grid (1.4,1.4);
\draw (0,0) parabola (1,1.5) parabola[bend at end] (2,0);
\draw (0,0) sin (1,1) cos (2,0) sin (3,-1) cos (4,0) sin (5,1);
```

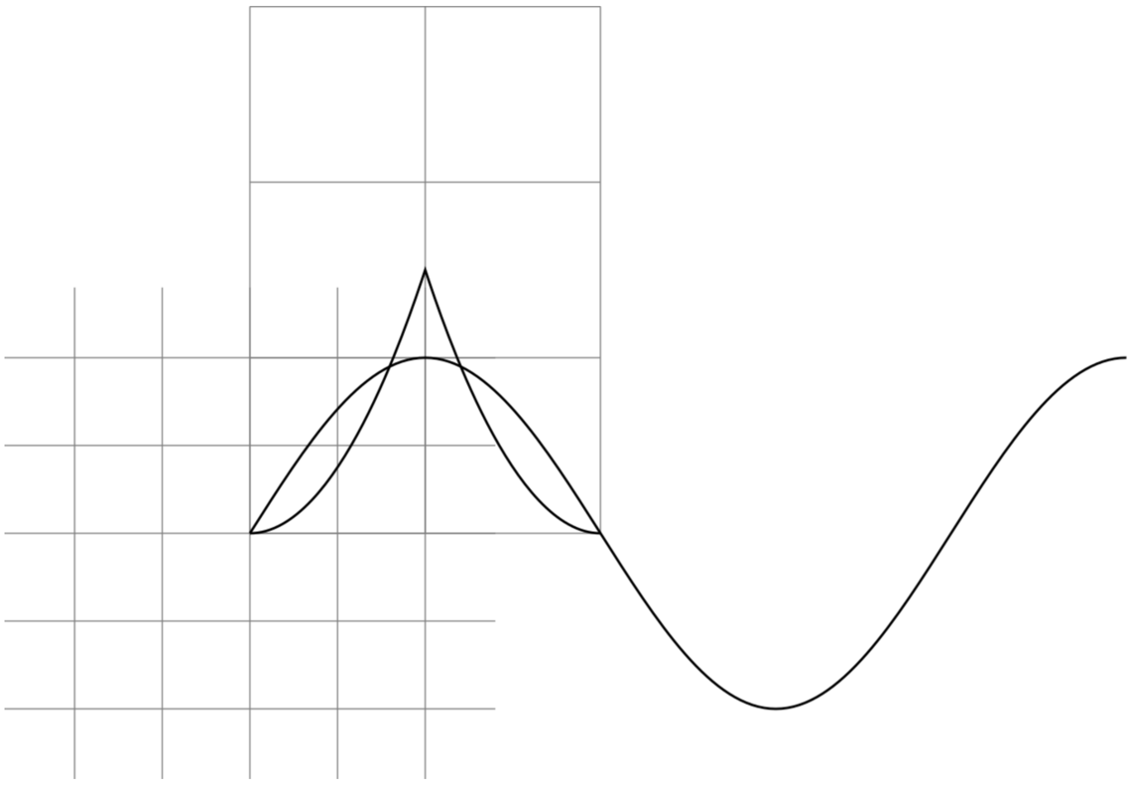


Figure 183

The option "help lines" denotes "fine gray".
To add arrow tips there are simple options for the drawing command:

```
\draw [->] (0,0) -- (30:20pt);
\draw [<->] (1,0) arc (180:30:10pt);
\draw [<<->] (2,0) -- ++(0.5,10pt) -- ++(0.5,-10pt) -- ++(0.5,10pt);
```

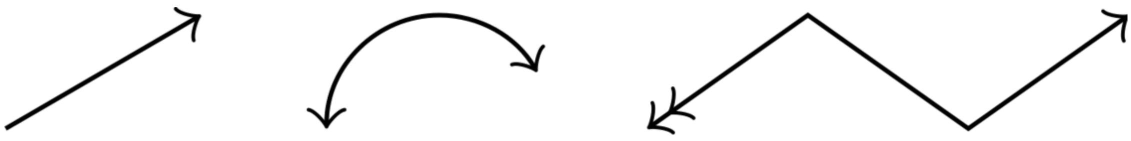


Figure 184

A loop can be realized by "`\foreach` ⟨`variable`⟩`in {`⟨`list of values`⟩`}` ⟨`commands`⟩ ".

```
\foreach \x in {0,...,9}
  \draw (\x,0) circle (0.4);
```

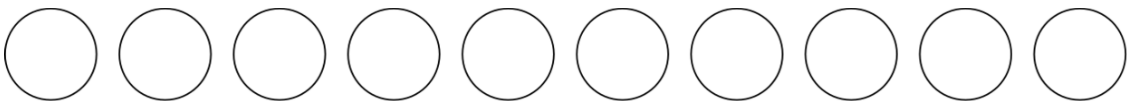


Figure 185

PGF also has a math engine which enables you to plot functions:

```
\draw [domain=<xmin>:<xmax>] plot (\x, {function});
```

Many functions are possible, here a selection: factorial(\x), sqrt(\x), pow(\x,y), exp(\x), ln(\x), log10(\x), log2(\x), abs(\x), mod(\x,y), round(\x), floor(\x), ceil(\x), sin(\x), cos(\x), tan(x), min(\x,y,), max(\x,y). The trigonometric functions assume that x is in degrees; if x is expressed in radians use e.g. sin(\x r). Two constants can be useful: e, which is equal to 2.718281828, and pi, which is equal to 3.141592654.
An example with two functions:

```
\draw [help lines] (-2,0) grid (2,4);
\draw [->] (-2.2,0) -- (2.2,0);
\draw [->] (0,0) -- (0,4.2);
\draw [green, thick, domain=-2:2] plot (\x, {4-\x*\x});
\draw [domain=-2:2, samples=50] plot (\x, {1+cos(pi*\x r});
```

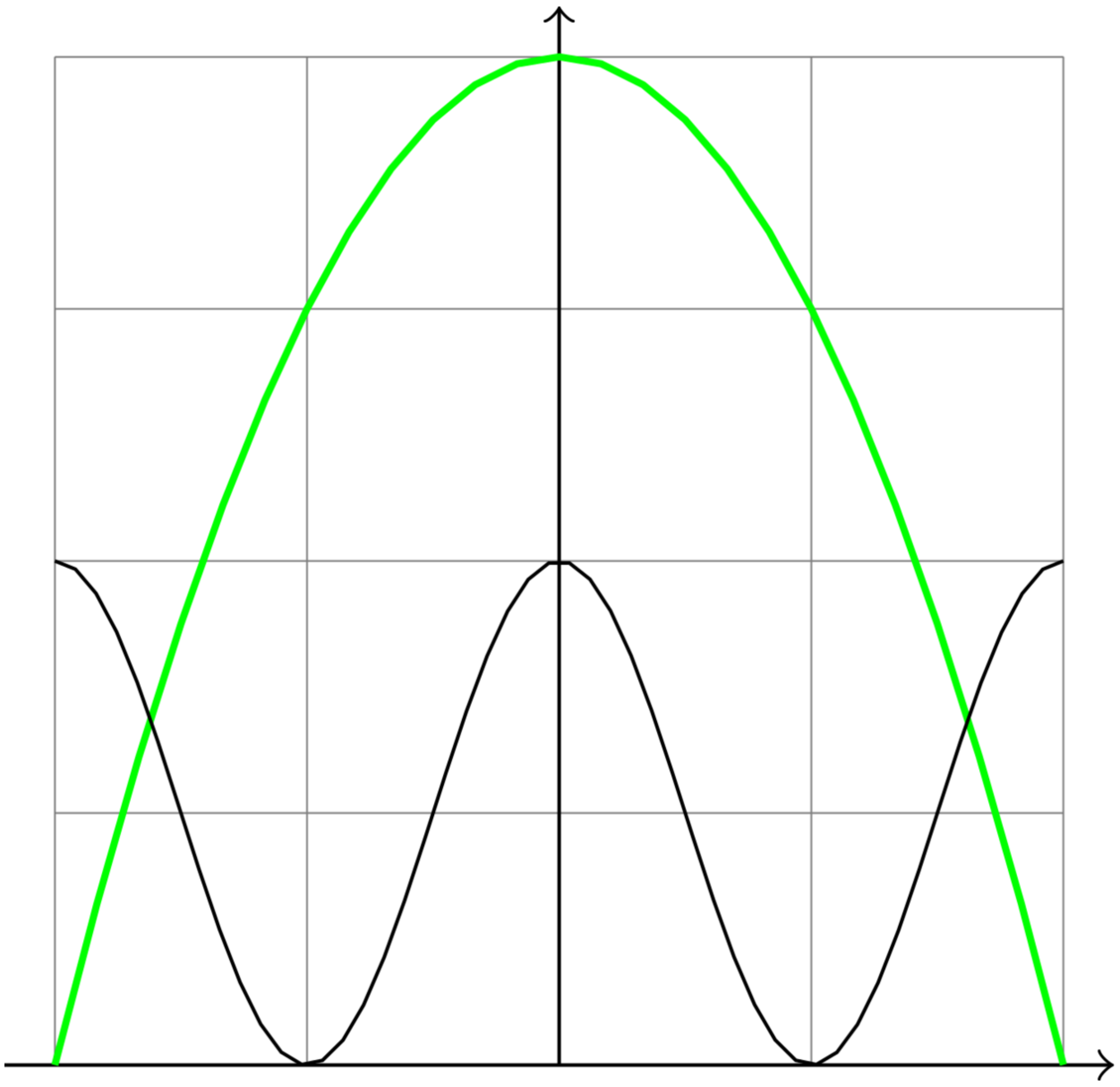


Figure 186

47.4. Nodes

A node is typically a rectangle or circle or another simple shape with some text on it. In the simplest case, a node is just some text that is placed at some coordinate. Nodes are not part of the path itself, they are added to the picture after the path has been drawn. Inside a path operation use the following syntax after a given coordinate:

```
node[<options>](<name>){<text>}
```

The "`(<name>)` " is a name for later reference and it is optional. If you only want to name a certain position without writing text there are two possibilities:

```
node[<options>](<name>){}
coordinate[<options>](<name>)
```

Writing text along a given path using the node command is shown as simple example:

```
\draw[dotted]
    (0,0) node {1st node}
 -- (1,1) node {2nd node}
 -- (0,2) node {3rd node}
 -- cycle;
```

Possible options for the node command are e.g. "`inner sep=<dimension>` ", "`outer sep=<dimension>` ", "`minimum size=<dimension>` ", "`shape aspect=<aspect ratio>` ", "`text=<color>` ", "`font=` ", "`align=<left_right_center>` ".
A node is centered at the current coordinate by default. Often it would be better to have the node to the besides the actual coordinate: Right ("`right` " or "`anchor=west` "), left ("`left` " or "`anchor=east` "), above ("`above` " or "`anchor=south` "), below ("`below` " or "`anchor=north` "). Combinations are also possible, like "`anchor=north east` " or "`below left` ".

```
\fill[fill=yellow]
    (0,0) node {1st node}
 -- (1,1) node[circle,inner sep=0pt,draw] {2nd node}
 -- (0,2) node[fill=red!20,draw,double,rounded corners] {3rd node};
```

To place nodes on a line or a curve use the "`pos=<fraction>` " option, where fraction is a floating point number between 0 representing the previous coordinate and 1 representing the current coordinate.

```
\draw (0,0) -- (3,1)
  node[pos=0]{0} node[pos=0.5]{1/2} node[pos=0.9]{9/10};
```

There exist some abbreviations: "`at start` " for "`pos=0` ", "`very near start` " for "`pos=0.125` ", "`near start` " for "`pos=0.25` ", "`midway` " for "`pos=0.5` ", "`near end` " for "`pos=0.75` ", "`very near end` " for "`pos=0.875` ", "`at end` " for "`pos=1` ".
The "`sloped` " option causes the node to be rotated to become a tangent to the curve.
Since nodes are often the only path operation on paths, there are special commands for creating paths containing only a node, the first with text ouput, the second without:

```
\node[<options>](<name>) at (<coordinate>){<text>};
\coordinate[<options>](<name>) at (<coordinate>);
```

One can connect nodes using the nodes' labels as coordinates. Having "`\path(0,0) node(x) {} (3,1) node(y) {};` " defined, the node at (0,0) got the name "`(x)` " and the one at (3,1) got a label "`(y)` ".

```
\path (0,0) node(x) {}
      (3,1) node(y) {};
\draw (x) -- (y);
```

Equivalent to

```
\coordinate (x) at (0,0);
\coordinate (y) at (3,1);
\draw (x) -- (y);
```

Multiline text can be included inside a node. A new line is indicated by double backslash "\\", but additionally you have to specify the alignment using the node option "align=". Here an example:

```
\filldraw
(0,0) circle (2pt) node[align=left,   below] {test 1\\is aligned left} --
(4,0) circle (2pt) node[align=center, below] {test 2\\is centered}     --
(8,0) circle (2pt) node[align=right,  below] {test 3\\is right aligned};
```

Path construction operations try to be clever, such that the path starts at the border of the node's shape and not from the node's center.

```
\path (0,0) node(x) {Hello World!}
(3,1) node[circle,draw](y) {$\int_1^2 x \mathrm d x$};
\draw[->,blue] (x) -- (y);
\draw[->,red] (x) -| node[near start,below] {label} (y);
\draw[->,orange] (x) .. controls +(up:1cm) and +(left:1cm) .. node[above,sloped]
 {label} (y);
```

Once the node x has been defined, you can use anchors as defined above relative to (x) as "`(x.<anchor>)` ", like "`(x.north)` ".

47.5. Examples

Example 1

```
\documentclass{article}
\usepackage{tikz}
\begin{document}
  \begin{tikzpicture}
  \draw[thick,rounded corners=8pt] (0,0) -- (0,2) -- (1,3.25)
   -- (2,2) -- (2,0) -- (0,2) -- (2,2) -- (0,0) -- (2,0);
  \end{tikzpicture}
\end{document}
```

Example 2

```
\documentclass{article}
\usepackage{tikz}
\begin{document}
 \begin{tikzpicture}[scale=3]
 \draw[step=.5cm, gray, very thin] (-1.2,-1.2) grid (1.2,1.2);
 \filldraw[fill=green!20,draw=green!50!black] (0,0) -- (3mm,0mm) arc (0:30:3mm)
 -- cycle;
 \draw[->] (-1.25,0) -- (1.25,0) coordinate (x axis);
 \draw[->] (0,-1.25) -- (0,1.25) coordinate (y axis);
 \draw (0,0) circle (1cm);
 \draw[very thick,red] (30:1cm) -- node[left,fill=white] {$\sin \alpha$} (30:1cm
 |- x axis);
 \draw[very thick,blue] (30:1cm |- x axis) -- node[below=2pt,fill=white] {$\cos
 \alpha$} (0,0);
 \draw (0,0) -- (30:1cm);
 \foreach \x/\xtext in {-1, -0.5/-\frac{1}{2}, 1}
   \draw (\x cm,1pt) -- (\x cm,-1pt) node[anchor=north,fill=white] {$\xtext$};
 \foreach \y/\ytext in {-1, -0.5/-\frac{1}{2}, 0.5/\frac{1}{2}, 1}
   \draw (1pt,\y cm) -- (-1pt,\y cm) node[anchor=east,fill=white] {$\ytext$};
 \end{tikzpicture}
\end{document}
```

Example 3: A Torus

```
\documentclass{article}
```

```
\usepackage{tikz}
\begin{document}
 \begin{tikzpicture}

        \draw (-1,0) to[bend left] (1,0);
        \draw (-1.2,.1) to[bend right] (1.2,.1);
        \draw[rotate=0] (0,0) ellipse (100pt and 50pt);

\end{tikzpicture}
\end{document}
```

Example 4: Some functions

```
\documentclass{article}
\usepackage{tikz}
\begin{document}
  \begin{tikzpicture}[domain=0:4]
    \draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
    \draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
    \draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};
    \draw[color=red]    plot (\x,\x)             node[right] {$f(x) =x$};
    \draw[color=blue]   plot (\x,{sin(\x r)})    node[right] {$f(x) = \sin x$};
    \draw[color=orange] plot (\x,{0.05*exp(\x)}) node[right] {$f(x) =
 \frac{1}{20} \mathrm e^x$};
  \end{tikzpicture}
\end{document}
```

48. PSTricks

PSTricks is a set of extensions. The base package is `pstricks` , other packages may be loaded when required.
The `xcolor` extension gets loaded along PSTricks, so there is no need to load it manually.
PSTricks has one technical specification: it uses PostScript internally, hence the name. Thus you cannot use the `pdftex` or `pdflatex` compilers, you will need to use `dvips` to get your proper document. It is still possible to get PDF from PS files thanks to `ps2pdf` . There is also the possibility to use the PDFTricks extension, which makes it feasible to use `pdflatex` together with PSTricks commands.
However, if you have installed the package `xetex-pstricks` , you can use `pstricks` with `xetex` or `xelatex` without modification of source file.

48.1. The `pspicture` environment

PSTricks commands are usually placed in a

```
pspicture
```

environment.

```
\begin{pspicture}(x1,y1)
% ...
\end{pspicture}
```

The first argument between parentheses specifies the coordinates of the upper-right corner of the picture. The bottom-left corner is at (0,0) and is placed at the reference point of the next character in the LaTeX document.
It is also possible to specify the coordinates (x0,y0) of the bottom-left corner:

```
\begin{pspicture}(x0,y0)(x1,y1)
% ...
\end{pspicture}
```

Thus the size of the picture is *(x1-x0)x(y1-y0)* . The default unit for coordinates is centimeters (cm); this can be changed with

```
\psset
```

, as in

```
\psset{unit=1bp}
```

. Any TeX dimension is allowed.

48.2. Fundamental objects

48.2.1. Lines and polylines

A simple line gets printed with

```
\psline(x0,y0)(x1,y1)
```

To get a vector, add an arrow as parameter:

```
\psline{->}(x0,y0)(x1,y1)
```

You can add as many points as you want to get a polyline:

```
\psline(x0,y0)(x1,y1)(x2,y3)…(xn,yn)
```

To get rounded corners, add the following option:

```
\psline[linearc=0.2,->](0,0)(0.5,0.5)(1,1)
```

48.2.2. Rectangles

```
\psframe(x0,y0)(x1,y1)
\psframe*(x0,y0)(x1,y1)
```

The starred version prints a filled rectangle. Use the following parameter to get rounded corners:

```
\psframe[framearc=0.2](x0,y0)(x1,y1)
```

48.2.3. Polygons

Polygons are always closed. The syntax is the same as for

```
\psline
```

:

```
\pspolygon(x0,y0)(x1,y1)(x2,y2)...(xn,yn)
```

As for rectangles, the starred version prints a filled polygon. And the

```
linearc=0.2
```

option will print rounded corners.

48.2.4. Circles, arc and ellipses

Starred version fills the shape.
For circles, you need to provide center coordinates and radius:

```
\pscircle(x,y){r}
```

To restrict the drawing to an arc, append the starting and ending angles in trigonometric notation:

```
\psarc(x,y){r}{angle1}{angle2}
```

Finally, ellipses:

```
\psellipse(x,y)(horizontal_axis,vertical_axis)
```

48.2.5. Curves

```
\psparabola(x0,y0)(x1,y1)
```

will print a symetric parabola with vertical asymptote, vertex *(x1,y1)* and ending at (x0,y0).
Use

```
\psbezier
```

to print a Bézier curve with an arbitrary number of control points. Arcs have at most 4 control points. Use the

```
showpoints=true
```
option to print the control points and the tangents.
Use

```
\pscurve
```
to print the interpolation of the given points. The

```
\psecurve
```
command omits the first and the last arcs.

48.3. Text

Use

```
\rput(x,y){text}
```
to print text. Provide an angle to rotate the text.

```
\rput{angle}(x,y){text}
```
You can provide the anchor of the text which will be at the specified coordinate.

```
\rput[t]{45}(5,5){text}
```
Available anchors:
- B, Bl, Br: baseline center, left and right.
- t, tl, tr: top center, left and right.
- b, bl, br: bottom center, left and right.

There is also the

```
\uput
```
command with further options:

```
\uput{distance}[angle](x,y){text}
```
The

```
distance
```
parameter is the distance from the coordinate.
PSTricks features several frame style for text.
- \psframebox{text}: rectangle.
- \psdblframebox{text}: double rectangle.
- \psshadowbox{text}: shaded rectangle.
- \pstcirclebox{text}: circle.
- \psovalbox{text}: oval.
- \psdiabox{text}: diamond.
- \pstribox{text}: triangle.

Example:

```
\rput(5,5){\psdiabox*[fillcolor=green]{text}}
```
Using the

```
pst-text
```
extension, it is possible to draw a text path.

```
\pstextpath{shape}{text}
```
To print a text following a path without printing the path, you need to use

```
\psset{linestyle=none}
```

.
Example:

```
\usepackage{pst-text}

% ...
\begin{pspicture}(5,5)
\psset{linestyle=none}
\pstextpath{\psline(0,0)(1,1)(2,0)}{triangle text}
\end{pspicture}
```

48.4. Grids

Without any parameter, the

`\psgrid`

command will print a grid all over the *pspicture* , with a spacing of 0.2 (*i.e.* 2mm). You can specify parameters:

- `\psgrid(xmax,ymax)`
  : prints a grid from *(0,0)* to *(xmax,ymax)* .
- `\psgrid(xmin,ymin)(xmax,ymax)`
  : prints a grid from *(xmin,ymin)* to *(xmax,ymax)* .
- `\psgrid(x0,y0)(xmin,ymin)(xmax,ymax)`
  : prints a grid from *(xmin,ymin)* to *(xmax,ymax)* , one of the node is at *(x0,y0)* .
- `griddots=value`
  : the full line of the main graduations is replaced by a dotted line. The *value* is the number of dots per graduation.
- `subgriddots=value`
  : same as

  `griddots`

  but for sub-graduations.
- `gridcolor=color,subgridcolor=color`
  : color of graduations and sub-graduations.
- `gridwidth=value,subgridwidth=value`
  : width of the lines.
- `subgriddiv=value`
  : number of subgraduations between two main graduations.
- `gridlabels=value`
  : size of the label numbers.
- `ticksize=value`
  : self-explanatory.
- `ticksize=valueneg valuepos`
  : same as above, but *valueneg* specifies the size for negative coordinates, *valuepos* for positive coordinates.
- `ticklinestyle=value`
  : self-explanatory. *value* may be one of

`solid, dashed, dotted`
. This is useful for huge graduations (*i.e.*

`ticksize`
is high).

Example

```
\psgrid[griddots=5, subgriddiv=0, gridlabels=0pt](-1,-1)(5,5)
```

Axis

If you want to add axes, use the `pstricks-add` extension with the following commands:

```
\psaxes(xmin,ymin)(xmax,ymax)
\psaxes(x0,y0)(xmin,ymin)(xmax,ymax)
```

(xmin,ymin) and *(xmax,ymax)* being the extreme, *(x0,y0)* being the intersection.

Options

- `Dx=value`
  and

  `Dy=value`
  defines the spacing between graduations.
- `comma`
  lets you use the comma as decimal separator.
- As for lines,

  `{->}`
  adds arrows on axes.

Example

```
\usepackage{pstricks-add}
% ...
\begin{pspicture}(-1,-1)(5,5)
\psaxes[comma,Dx=0.5,Dy=0.5]{->}(0,0)(3,3)
\end{pspicture}
```

48.5. Generic parameters

48.5.1. All shapes

These are to be added between square brackets.

- `linewidth=value`
  : if *value* is without unit, then the default unit is used.
- `linecolor=color`
  : *color* is as defined by the `xcolor` package.
- `linestyle=value`
  : *value* is one of

  `dashed,dotted`

  .
- `doubleline=true`

  .
- `showpoints=true`

: highlights points.

-
  ```
  dotscale=value
  ```
  specifies the size of the points.
-
  ```
  dotstyle=value
  ```
  where *value* is among:
  - *: disc
  - o: circle
  - +,x: cross
  - square, squarre*: starred version is filled.
  - diamond, diamond*
  - triangle, triangle*
  - etc.

For example

```
\pscircle[linewidth=0.2,linestyle=dashed,linecolor=blue](0,0){1}
```

To apply parameters globally:

```
\psset{linewidth=0.2,linestyle=dashed,linecolor=blue}
\pscircle(0,0){1}
```

This command also lets you change the default unit for lengths.

-
  ```
  unit=value
  ```
-
  ```
  xunit=value
  ```
  and

  ```
  yunit=value
  ```

value is a number with or without unit. This changes the scale of the drawings, but will not change the width of lines.

48.5.2. Open shapes

You can define the extreme of an open shape (line, polyline, arc, etc.) with an optional parameter

```
{symbol1-symbol2}
```

. There is a decent list of available symbols.

- < or >: arrow.
- << or >>: double arrow.
- |: bar.
- |*: centered bar.
- oo: circle.
- o: centered circle.
- **: disk.
- *: centered disk.
- |< or >|: arrow plus bar.
- cc: rounded extreme.
- c: centered rounded extreme.

Example:

```
\psline{|->>}(x0,y0)(x1,y1)
```

48.5.3. Closed shapes

For closed shape you may define the fillstyle.

- `fillstyle=value`
  : pattern. Possible values:

  `crosshatch, crosshatch*, vlines, vlines*, hlines, hlines*, solid`

  .
- `fillcolor=color`

  .
- `hatchcolor=color`

  .
- `hatchwidth=value`

  .
- `hatchsep=value`

  .
- `hatchangle=value`

  .

Example:

```
\pscircle[hatchcolor=blue,fillstyle=vlines](0,0){1}
```

48.6. Object location

The

`\rput`
and

`\uput`
commands can be used to move any object.
Example

```
\begin{pspicture}(5,5)
\psline{->}(0,0)(1,1)
\rput(1,1){\psline{->}(0,0)(1,1)}
\end{pspicture}
```

or

```
\begin{pspicture}(5,5)
\psline{->}(0,0)(1,1)
\psline{->}(1,1)(2,2)
\end{pspicture}
```

You can repeat the operation with

`\multirput`

:

```
\multirput(x0,y0)(xoffset, yoffset){times}{object}
```

You can use the same options as for

`\rput`

:

```
\multirput[reference]{angle}(x0,y0)(xoffset, yoffset){times}{object}
```

With no text but with graphics only, you can use the

```
\multips
```

command:

```
\multips(x0,y0)(xoffset, yoffset){times}{object}
\multips{angle}(x0,y0)(xoffset,yoffset){times}{object}
```

48.7. The `PDFTricks` extension

The original `PSTricks` package does not work with `pdflatex` , but thankfully `PDFTricks` allows us to bypass this limitation.

Usage

- Declare the `PDFTricks` packages in the preamble.
- Place all `PSTricks` extensions in a

  ```
  psinputs
  ```

  environment; place all `PSTricks` commands in a

  ```
  pdfpic
  ```

  environment.
- Compile with `pdflatex -shell-escape <file>` .

The `-shell-escape` parameter enables shell command calls. It is required for `PDFTricks` to run.

Example

```
\documentclass{article}

\usepackage{pdftricks}
\begin{psinputs}
   \usepackage{pstricks}
   \usepackage{multido}
\end{psinputs}

% ...

\begin{document}

% ...

\begin{pdfpic}
   \psset{unit=\linewidth}
   \begin{pspicture}(0,0)(10,10)
      [...]
   \end{pspicture}
\end{pdfpic}

% ...

\end{document}
```

Another way to use `PSTricks` with `pdflatex` is the `pst-pdf` package.

49. Xy-pic

xy is a special package for drawing diagrams. To use it, simply add the following line to the preamble of your document:

```
\usepackage[all]{xy}
```

where "all" means you want to load a large standard set of functions from *Xy-pic* , suitable for developing the kind of diagrams discussed here.
The primary way to draw *Xy-pic* diagrams is over a matrix-oriented canvas, where each diagram element is placed in a matrix slot:

```
\begin{displaymath}
    \xymatrix{A & B \\
              C & D }
\end{displaymath}
```

$$\begin{tikzcd}
A & B \\
C & D
\end{tikzcd}$$

Figure 187

The `\xymatrix` command must be used in math mode. Here, we specified two lines and two columns. To make this matrix a diagram we just add directed arrows using the `\ar` command.

```
\begin{displaymath}
    \xymatrix{ A \ar[r] & B \ar[d] \\
               D \ar[u] & C \ar[l] }
\end{displaymath}
```

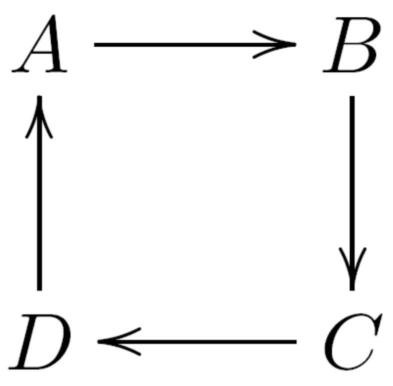

$$\begin{tikzcd} A \arrow[r] & B \arrow[d] \\ D \arrow[u] & C \arrow[l] \end{tikzcd}$$

Figure 188

The arrow command is placed on the origin cell for the arrow. The arguments are the direction the arrow should point to (up, down, right and left).

```
\begin{displaymath}
    \xymatrix{
        A \ar[d] \ar[dr] \ar[r] & B \\
        D                       & C }
\end{displaymath}
```

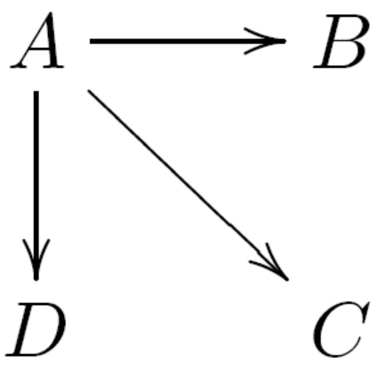
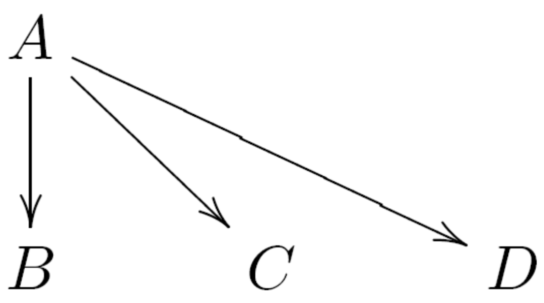

$$\begin{tikzcd} A \arrow[r] \arrow[d] \arrow[dr] & B \\ D & C \end{tikzcd}$$

Figure 189

To make diagonals, just use more than one direction. In fact, you can repeat directions to make bigger arrows.

```
\begin{displaymath}
    \xymatrix{
        A \ar[d] \ar[dr] \ar[drr] &   &   \\
        B                         & C & D }
\end{displaymath}
```

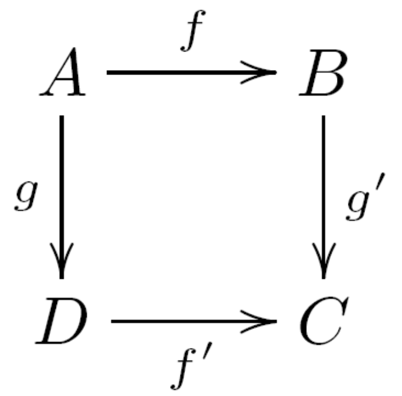


$$\begin{tikzcd}
A \arrow[d] \arrow[r] \arrow[rr] & & \\
B & C & D
\end{tikzcd}$$

Figure 190

We can draw even more interesting diagrams by adding labels to the arrows. To do this, we use the common superscript and subscript operators.

```
\begin{displaymath}
    \xymatrix{
        A \ar[r]^f \ar[d]_g & B \ar[d]^{g'} \\
        D \ar[r]_{f'}       & C }
\end{displaymath}
```



$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "g'"] \\
D \arrow[r, "f'"'] & C
\end{tikzcd}$$

Figure 191

As shown, you use these operators as in math mode. The only difference is that that superscript means ”on top of the arrow”, and subscript means ”under the arrow”. There is a third operator, the vertical bar: | It causes text to be placed in the arrow.

```
\begin{displaymath}
    \xymatrix{
        A \ar[r]f \ar[d]g & B \ar[d]{g'} \\
        D \ar[r]{f'}      & C }
\end{displaymath}
```

$$\begin{tikzcd}
A \arrow[r, "f" description] \arrow[d, "g" description] & B \arrow[d, "g'" description] \\
D \arrow[r, "f'" description] & C
\end{tikzcd}$$

Figure 192

To draw an arrow with a hole in it, use `\ar[...]|\hole` . In some situations, it is important to distinguish between different types of arrows. This can be done by putting labels on them, or changing their appearance

```
\shorthandoff{"}
\begin{displaymath}
    \xymatrix{
        \bullet\ar@{->}[rr]     && \bullet\\
        \bullet\ar@{.<}[rr]     && \bullet\\
        \bullet\ar@{~)}[rr]     && \bullet\\
        \bullet\ar@{=(}[rr]     && \bullet\\
        \bullet\ar@{~/}[rr]     && \bullet\\
        \bullet\ar@{^{(}->}[rr] && \bullet\\
        \bullet\ar@2{->}[rr]    && \bullet\\
        \bullet\ar@3{->}[rr]    && \bullet\\
        \bullet\ar@{=+}[rr]     && \bullet }
\end{displaymath}
\shorthandon{"}
```

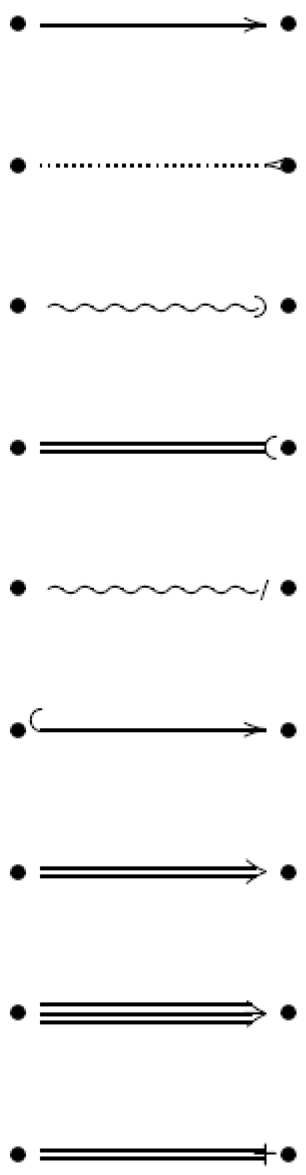


Figure 193

Notice the difference between the following two diagrams:

```
\begin{displaymath}
    \xymatrix{ \bullet \ar[r] \ar@{.>}[r] & \bullet }
\end{displaymath}
```

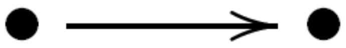


Figure 194

```
\begin{displaymath}
    \xymatrix{
        \bullet \ar@/^/[r]
        \ar@/_/@{.>}[r] &
        \bullet }
\end{displaymath}
```

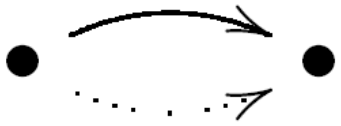


Figure 195

The modifiers between the slashes define how the curves are drawn. *Xy-pic* offers many ways to influence the drawing of curves; for more information, check the *Xy-pic* documentation.
If you are interested in a more thorough introduction then consult the Xy-pic Home Page[1], which contains links to several other tutorials as well as the reference documentation.

1 `http://xy-pic.sourceforge.net`

50. Creating 3D graphics

For creating three-dimensional graphics, there is basic functionality in the PGF/TikZ[1] package, although drawing 3D graphics with PGF/TikZ is very non-flexible, mainly because it lacks functionality for identifying the surfaces that are covered by other surfaces and should be excluded from the rendered image.
A package that can handle this correctly is the `pst-solides3d` [2] package.
Another way to create 3D graphics is to use Asymptote[3]

1 Chapter 47 on page 535
2 `http://www.ctan.org/pkg/pst-solides3d`
3 `http://en.wikipedia.org/wiki/Asymptote%20%28vector%20graphics%20language%29`

Part VIII.

Programming

51. Macros

Documents produced with the commands you have learned up to this point will look acceptable to a large audience. While they are not fancy-looking, 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 that matches your needs, or the output produced by some existing command may not meet your requirements.
In this chapter, we will try to give some hints on how to teach LaTeX new tricks and how to make it produce output that looks different from what is provided by default.
LaTeX is a fairly high-level language compared to Plain TeX and thus is more limited. The next chapter[1] will focus on Plain TeX and will explain advanced techniques for programming.

51.1. 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. Note that the command *name* can but need not be enclosed in braces, as you like. 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 `\wbal` that will print "The Wikibook about LaTeX". Such a command could come in handy if you had to write the title of this book over and over again.

```
\newcommand{\wbal}{The Wikibook about \LaTeX}
This is ‘‘\wbal
```

This is “The Wikibook about LaTeX” ... “The Wikibook about LaTeX”

The next example illustrates how to define a new command that 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, these arguments are added in an extra set of brackets.

1 Chapter 52 on page 579

```
\newcommand{\wbalsup}[1] {
  This is the Wikibook about LaTeX
  supported by #1}
\newcommand{\wbalTwo}[2] {
  This is the Wikibook about LaTeX
  supported by #1 and #2}
% in the document body:
\begin{itemize}
\item \wbalsup{Wikimedia}
\item \wbalsup{lots of users!}
\item \wbalTwo{John Doe}{Anthea Smith}
\end{itemize}
```

- This is the Wikibook about LaTeX supported by Wikimedia
- This is the Wikibook about LaTeX supported by lots of users!
- This is the Wikibook about LaTeX supported by John Doe and Anthea Smith

Name your new command `\wbalTwo` and not `\wbal2` as digits cannot be used to name macros — invalid characters will error out at compile-time.
LaTeX will not allow you to create a new command that 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 will silently ignore the new command.
With LaTex2e, it is also possible to add a default parameter to a command with the following syntax:

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

If the default parameter of `\newcommand` is present, then the first of the number of arguments specified by `num` is optional with a default value of `default`; if absent, then all of the arguments are required.

```
\newcommand{\wbalTwo}[2][Wikimedia]{
  This is the Wikibook about LaTeX
  supported by {#1} and {#2}!}
% in the document body:
\begin{itemize}
\item \wbalTwo{John Doe}
\item \wbalTwo[lots of users]{John Doe}
\end{itemize}
```

- This is the Wikibook about LaTeX supported by Wikimedia and John Doe!
- This is the Wikibook about LaTeX supported by lots of users and John Doe!

Note

When the command is used with an explicit first parameter it is given enclosed with brackets (here "`[lots of users]`").

Here is a common example: if you are writing a book about Mathematics and you have to use vectors, you have to decide how they will look. There are several different standards, used in many books. If a is a vector, some people like to add an arrow over it ($\vec{a}$), other people write it underlined ($\underline{a}$); another common version is to write it bold (**a**). Let us assume you want to write your vectors with an arrow over them; then add the following line in your `mystyle.sty` .

```
\newcommand{\myvec}[1]{\vec{#1}}
```

and write your vectors inside the new `\myvec{...}` command. You can call it as you wish, but you'd better choose a short name because you will probably write it very often. Then, if you change your mind and you want your vectors to look differently you just have to change the definition of your `\myvec{...}`. Use this approach whenever you can: this will save you a lot of time and increase the consistency of your document.

51.1.1. DeclareRobustCommand

Some commands are *fragile* , that is they fail in some environments. If a macro works in body text but not in (for example) a figure caption, it's worth trying to replace the `\newcommand{\MyCommand}...` declaration with `\DeclareRobustCommand{\MyCommand}...` in the preamble. This is especially true for macros which, when expanded, produce text that is written to a `.aux` file.

51.2. New environments

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

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

Again `\newenvironment` can have an optional argument. When the `\begin{name}` command (which starts the environment) is encountered, 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 (which ends the environment) is encountered.

The `num` argument is used the same way as in the `\newcommand` command. LaTeX makes sure that you do not define an environment that 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 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}
```

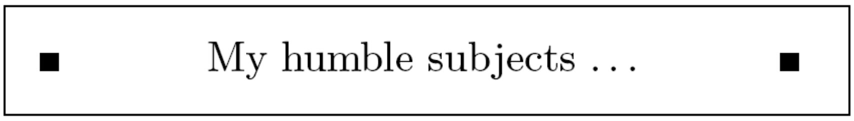


Figure 196

51.2.1. Unmatched braces

Often, part of the motive behind creating an environment is to surround its body in a grouping of braces. However, this requires unmatched braces to appear in both the beginning and end portions of the environment declaration, which will prevent the document from compiling. To solve this issue, use the TeX synonyms `\bgroup` and `\egroup` instead of typing `{` and `}` in this case.

51.2.2. Extra space

When creating a new environment you may easily get bitten by extra spaces creeping in, which can potentially have fatal effects. For example when you want to create a title environment which suppresses its own indentation as well as the one on the following paragraph. The `\ignorespaces` command in the begin block of the environment will make it ignore any space after executing the begin block. The end block is a bit more tricky as special processing occurs at the end of an environment. With the `\ignorespacesafterend` LaTeX will issue an `\ignorespaces` after the special 'end' processing has occurred.

```
\newenvironment{simple}%
{\noindent}%
{\par\noindent}

\begin{simple}
See the space\\to the left.
\end{simple}
Same\\here.
```

```
  See the space
to the left.

  Same
here.
```

```
\newenvironment{correct}%
{\noindent\ignorespaces}%
{\par\noindent%
\ignorespacesafterend}

\begin{correct}
No space\\to the left.
\end{correct}
Same\\here.
```

```
No space
to the left.

Same
here.
```

Also, if you're still having problems with extra space being appended at the end of your environment when using the `\input` for external source, make sure there is no space between the beginning, sourcing, and end of the environment, such as:

```
\begin{correct}\input{somefile.tex}\end{correct}
```

or

```
\begin{correct}%
\input{somefile.tex}%
\end{correct}
```

51.3. Declare commands within newenvironment

New commands can be declared within newenvironment. Commands declared within the newenvironment refer to their arguments by doubling the # character. In the following example, a new environment is declared along with a nested command:

```
\newenvironment{topics}{
\newcommand{\topic}[2]{ \item{##1 / ##2\} }
Topics:
\begin{itemize}
}
{
\end{itemize}
}
```

If, by mistake, the arguments passed to the \topics macro are defined with a single # character, the following error message will be thrown:

```
! Illegal parameter number in definition of \topics.
```

51.4. Extending the number of arguments

The `xkeyval` packages will let you define key/value options for commands.

```
\mycommand[key1=value1, key3=value3]{some text}
```

The package is quite complete and documentation is exhaustive. We recommend that package developers read it. `http://www.ctan.org/pkg/xkeyval`
Let's provide a simple example[2]:

```
\usepackage{xkeyval}
% ...

\makeatletter
\def\my@emphstyle#1{\csname my@style@#1\endcsname}
%% Predefined styles
\providecommand\my@style@default{\em}
\providecommand\my@style@bold{\bfseries}

\define@key{myemph}{code}{%
  \def\my@emphstyle{#1}
}
\define@key{myemph}{style}{%
  \def\my@emphstyle{\csname my@style@#1\endcsname}
}
\newcommand\setemph[1]{%
  \setkeys{myemph}{#1}
}

\renewcommand\emph[1]{%
  {\my@emphstyle #1}
}

\makeatother

Something \emph{important}

\setemph{style=bold}
Something \emph{important}

\setemph{code=\Large\sffamily}
Something \emph{important}
```

51.5. Arithmetic

UNKNOWN TEMPLATE Expand
LaTeX can manipulate numbers.
The `calc` package provides the common infix notation.

```
\usepackage{calc}
% ...
\newcounter{mine}
\setcounter{mine}{2*17}
\themine
```

For high-precision computations, you can use the `fp`[3] package.

```
\usepackage{fp}

% Clip
\[
\FPmul\result{2}{7}
\FPclip\result\result
2*7 = \result
\]
```

2 tex.stackexchange.com ^{`http://tex.stackexchange.com/questions/13270/a-package-template-using-xkeyval`}
3 ctan.mackichan.com ^{`http://ctan.mackichan.com/macros/latex/contrib/fp/README`}

```
% Infix
\[
\newcommand\result{11}
\sqrt{\sin(2+\result)} \approx
\FPeval\result{round(root(2,sin(result + 2.5)),2)}
\result
\]

% Postfix
\[
\FPupn\result{17 2.5 + 17.5 swap - 2 1 + * 2 swap /} % or \FPupn\result{2 17.5
 17 2.5 + - 2 1 + * /}
\FPclip\result\result
(17+2.5 - 17.5) * (2+1) / 2  = \result
\]

% High precision
\[
\FPdiv\result{17}{7}
\frac{17}{7} \approx \FPtrunc\result\result{3}
\result
\]
```

51.6. Conditionals

LaTeX can use conditionals thanks to the `ifthen` package.

```
\usepackage{ifthen}
% ...

\ifthenelse{ \equal{\myvar}{true} }{
  This is true.
}{
  This is false.
}
```

51.7. Loops

The `PGF/TikZ` extension provides the `\foreach` command.

```
\usepackage{tikz}
% ...

\foreach \i/\q in {wheat/50g, water/1L, yeast/2g}{
  \noindent\i\dotfill\q\\
}
```

If you are only using `\foreach` and not drawing graphics, you may instead use the `pgffor` package directly.
Alternatively you can check out the `multido` package.

51.8. Strings

`xstring` provides a lot of features. From CTAN:

- testing a string's contents
- extracting substrings
- substitution of substrings
- string length

- position of a substring
- number of recurrences of a substring

Examples:

```
\usepackage{xstring}
% ...

\newcommand\mystr{Hello World!}

The string ``\mystr'' has \StrLen{\mystr}{} characters.

Predicate ``\mystr{} contains the word Hello'' is
 \IfSubStr{\mystr}{Hello}{true}{false}.
```

51.9. LaTeX Hooks

LaTeX provide two hooks:

- `\AtBeginDocument` will let you specify a set of commands that will be executed when `\begin{document}` is met.
- `\AtEndDocument` does the same for `\end{document}`.

This gives you some more flexiblity for macros. It can be useful to override settings that get executed after the preamble. These hooks can be called several times. The commands will be executed in the order they were set.
For instance, let's replace the page numbers with oldstylenums:

```
\usepackage{textcomp}

\AtBeginDocument{%
  % Make the page numbers in text figures
  \let\myThePage\thepage
  \renewcommand{\thepage}{ \oldstylenums{\myThePage} }
}
```

There are also hooks for classes and packages. See Creating Packages[4].

51.10. Command-line LaTeX

If you work on a Unix-like OS, you might be using Makefiles or any kind of script to build your LaTeX projects. In that connection it might be interesting to produce different versions of the same document by calling LaTeX with command-line parameters. If you add the following structure to your document:

```
\usepackage{ifthen}
%...

% default value.
\providecommand\blackandwhite{false}
%...

\ifthenelse{ \equal{\blackandwhite}{true} }{
% "black and white" mode; do something..
}{
% "color" mode; do something different..
}
```

Now you can call LaTeX like this:

4 Chapter 53 on page 593

```
    latex '\providecommand{\blackandwhite}{true}\input{test.tex}'
```

First the command `\blackandwhite` gets defined and then the actual file is read with input. By setting `\blackandwhite` to false the color version of the document would be produced.

51.11. Notes and References

52. Plain TeX

While you play with LaTeX macros, you will notice that it is quite limited. You may wonder how all these packages you are using every day have been implemented with so little. In fact, LaTeX is a set of Plain TeX macros and most packages use Plain TeX code. Plain TeX is much more low-level, it has much more capabilities at the cost of a steep learning curve and complex programming.
Up to a few exceptions, you can use the full Plain TeX language within a valid LaTeX document whereas the opposite is false.

52.1. Vocabulary

To avoid confusion it seems necessary to explain some terms.

- A *group* is everything after an opening brace and before the matching closing brace.
- A *token* is a character, a control sequence, or a group.
- A *control sequence* is anything that begins with a \. It is not printed as is, it is expanded by the TeX engine according to its type.
- A *command* (or *function* or *macro*) is a control sequence that may expand to text, to (re)definition of control sequences, etc.
- A *primitive* is a command that is hard coded in the TeX engine, *i.e.* it is not written in Plain TeX.
- A *register* is the TeX way to handle variables. They are limited in numbers (256 for each type of register in classic TeX, 32767 in e-TeX).
- A *length* is a control sequence that contains a length (a number followed by a unit). See Lengths[1].
- A *font* is a control sequence that refers to a font file. See Fonts[2].
- A *box* is an object that is made for printing. Anything that ends on the paper is a box: letters, paragraphs, pages... See Boxes[3].
- A *glue* is a certain amount of space that is put between boxes when they are being concatenated.
- A *counter* is a register containing a number. See Counters[4].

There may be more terms, but we hope that it will do it for now.

52.2. Catcodes

In TeX some characters have a special meaning that is not to print the associated glyph. For example, \ is used to introduce a control sequence, and will not print a backslash by default.

1 Chapter 23 on page 283
2 Chapter 9 on page 95
3 Chapter 25 on page 295
4 Chapter 24 on page 291

To distinguish between different meanings of the characters, TeX split them into *category codes* , or *catcodes* for short. There are 16 category codes in TeX.
A powerful feature of TeX is its ability to redefine the language itself, since there is a `\catcode` function that will let you change the category code of any characters.
However, this is not recommended, as it can make code difficult to read. Should you redefine any catcode in a class or in a style file, make sure to revert it back at the end of your file.
If you redefine catcodes in your document, make sure to do it after the preamble to prevent clashes with package loading.

Code	**Description**	**Default set**
0	Escape character and control sequences	`\`
1	Beginning of group	`{`
2	End of group	`}`
3	Math shift	`$`
4	Alignment tab	`&`
5	End of line	`^^M` (ASCII return)
6	Macro parameter	`#`
7	Superscript	`^` and `^^K`
8	Subscript	`_` and `^^A`
9	Ignored character	`^^@` (ASCII null)
10	Space	␣ and `^^I` (ASCII horizontal tab)
11	Letter	`A...Z` and `a...z`
12	Other character	everything not listed in the other catcodes. Most notably, @.
13	Active character	`~` and `^^L` (ASCII form feed)
14	Comment character	`%`
15	Invalid character	`^^?` (ASCII delete)

52.2.1. Active characters

Active characters resemble macros: they are single characters that will expand before any other command.

```
\catcode` = 13
\def{\TeX}
...
This is a stupid example of .
```

This is a stupid example of TeX.

Note that an active character needs to be directly followed by a definition, otherwise the compilation will fail.

52.2.2. Examples

Texinfo

Texinfo[5] uses a syntax similar to TeX with one major difference: all functions are introduced with a @ instead of a \. This is not by chance: it actually uses TeX to print the PDF version of the files. What it basically does is inputting `texinfo.tex` which redefines the control sequence character. Possible implementation:

```
\catcode`\@=0
@def@@{@char64} % To write '@' character.
\catcode`\\=13 @def\{{@tt @char92}}

The @TeX command was previously written '\TeX', now it is written '@@TeX'.
```

The TeX command was previously written '\TeX' It is now written '@TeX'.

With this redefinition, the '@' should now introduce every command, while the '\' will actually print a backslash character.

Itemize

Some may find the LaTeX syntax of list environments a bit cumbersome. Here is a quick way to define a wiki-like itemize:

```
\catcode` = 13
\def{\item {--}}
\def\itemize#1{{\leftskip = 40 pt #1 \par}}

\itemize{
 First item
 Second item
}
```

Dollar and math

If you have many 'dollar' symbols to print, you may be better off to change the math shift character.

```
\catcode`$ = 11
\catcode` = 3

It costs $100.
Let's do the math: 50+50=100. Let's highlight it:
50+50=100
```

52.2.3. *\makeatletter* and *\makeatother*

If you have done a bit of LaTeX hacking, you must have encountered those two commands, `\makeatletter` and `\makeatother`.

In TeX the '@' characters belongs to catcode 11 *letters* by default. It means you can use it for macro names. LaTeX makes use of the catcode to specify a rule: all non-public, internal macros that are not supposed to be accessed by the end-user contains at least one '@' character in their name. In the document, LaTeX changes the catcode of '@' to 12, *others* .

5 `http://en.wikipedia.org/wiki/Texinfo`

That's why when you need to access LaTeX internals, you must enclose all the commands accessing private functions with `\makeatletter` and `\makeatother`. All they do is just changing the catcode:

```
\def\makeatletter{\catcode`@ = 11}
\def\makeatother{\catcode`@ = 12}
```

52.3. Plain TeX macros

`\newcommand` and `\renewcommand` are LaTeX-specific control sequences. They check that no existing command gets shadowed by the new definition.
In Plain TeX, the primitives for macro definition make no check on possible shadowing. It's up to you to make sure you are not breaking anything.
The syntax is

```
\def<macroname> #1<sep1>#2<sep2>{macro content, use of argument #1, blah, #2
 ...}
```

You can use (almost) any sequence of character between arguments. For instance let's write a simple macro that will convert the decimal separator from point to comma. First try:

```
\def\pointtocomma #1.#2{(#1,#2)}
%%...

\pointtocomma 123.456
```

This will print *(123,4)56* . We added the parentheses just to highlight the issue here. Each parameter is the shortest possible input sequence that matches the macro definition, separators included. Thus `#1` matches all characters up to the first point, and `#2` matches the first token only, *i.e.* the first character, since there is no separator after it.
Solution: add a second separator. A space may seem convenient:

```
\def\pointtocomma #1.#2 {(#1,#2)}
```

As a general rule, everytime you expect several parameters with specific separators, think out the last separator. If you do not want to play with separators, then Plain TeX macros are used just as LaTeX macros (without default parameter):

```
\def\mymacro#1#2#3{{\bf #1}#2{\bf #3}}
%% ...
\mymacro{word1}{word2 word3}{!!!}
```

52.3.1. Expanded definitions

TeX has another definition command: `\edef`, which stands for *expanded def*. The syntax remains the same:

```
\edef<macroname> <argumentslist>{<expanded content>}
```

The content gets expanded (but not executed, *i.e.* printed) at the point where `\edef` is used, instead of where the defined macro is used. Macro expansion is not always obvious...
Example:

```
\def\intro{Example}
\edef\example#1{\intro~---~#1}
\def\intro{Exercise}

\example{This is an example}
```

Here the redefinition of `\intro` will have no effect on `\example`.

52.3.2. Global definitions

Definitions are limited to their scope. However it might be convenient sometimes to define a macro inside a group that remain valid outside the group, and until the end of the document. This is what we call *global definitions* .

```
{
\def\LocalTeX{Local\TeX}
\global\def\GlobalTeX{Global\TeX}
}
I can still access the \GlobalTeX{} macro here.
```

You can also use the `\global` command with `\edef`.
Both commands have a shortcut:

- `\gdef` for `\global\def`
- `\xdef` for `\global\edef`

52.3.3. Long definitions

The previous definition commands would not allow you to use them over multiple paragraphs, *i.e.* text containing the `\par` command -- or double line breaks.
You can prefix the definition with the `\long` command to allow multi-paragraph arguments.
Example:

```
\long\def\dummy#1{#1}
\dummy{First paragraph\par Second paragraph}
```

52.3.4. Outer definitions

This prefix macro prevent definitions from being used in some context. It is useful to consolidate macros and make them less error-prone because of bad contexts. *Outer macros* are meant to be used outside of any context, hence the name.
For instance the following code will fail:

```
\outer\def\test{a test}
\def\failure{\test}
```

Outer macros are not allowed to appear in:

- macro parameters
- skipped conditional
- ...

52.3.5. *let* and *futurelet*

`\let<csname><token>` is the same as `\expandafter\def\expandafter<csname>\expandafter{<content>`
It defines a new control sequence name which is equivalent to the specified `token`. The `token` is usually another control sequence.
Note that `\let` will expand the `token` one time only, contrary to `\edef` which will expand recursively until no further expansion is possible.
Example[6]:

6 From tex.stackexchange.com ^{`http://tex.stackexchange.com/questions/8163/what-is-the-difference-between-let-and-edef`} : What is the difference between \let and \edef?

```
Using let:\par
\def\txt{a}
\def\foo{\txt}
\let\bar\foo
\bar % Prints a
\def\txt{b}
\bar % Prints b

Using edef:\par
\def\txt{a}
\def\foo{\txt}
\edef\bar{\foo}
\bar % Prints a
\def\txt{b}
\bar % Prints a
```

`\futurelet<csname><token1><token2>...` works a bit differently. `token2` is assigned to `csname`; after that TeX processes the `<token1><token2>...` sequence. So `\futurelet` allows you to assign a token while using it right after.

52.3.6. Special control sequence name

Some macros may have a name that is not directly writable as is. This is the case of macros whose name is made up of macro names. Example:

```
\def\status{full}
\def\varempty{This is empty}
\def\varfull{This is full}

\csname var\status \endcsname
```

The last line will print a sentence depending on the `\status`.
This command actually does the opposite of `\string` which prints a control sequence name without expanding it:

```
{\tt \string\TeX}
```

```
\TeX
```

52.3.7. Controlling expansion

`\expandafter{token1}{token2}` will expand `token2` before `token1`. It is sometimes needed when `token2` expansion is desired but cannot happen because of `token1`.

```
{\tt \expandafter\string\csname TeX\endcsname}
```

```
\TeX
```

`\noexpand` is useful to have fine grained control over what gets expanded in an `\edef`. Example:

```
\def\intro{Example}
\def\separator{~---~}
```

```
\edef\example#1{\intro\noexpand\separator#1}

\example{no expand makes the separator dynamic in an {\tt \string\edef}.}

\def\intro{For instance}
\def\separator{~:~}

\example{the separator changed, but not the first word.}
```

`\the` control sequence will let you see the content of various TeX types:

- catcodes
- chardef
- font parameters
- internal parameters
- lengths
- registers
- ...

Example:

```
Text dimensions: $ \the\hsize \times \the\vsize $
```

52.4. Registers

Registers are kind of typed variables. They are limited in numbers, ranging from 0 to 255. There are 6 different types:

Type	**Description**
box	one box
count	an integer
dimen	a length
muskip	a glue (in mu unit)
skip	a glue
toks	a sequence of tokens

TeX uses some registers internally, so you would be better off not using them.
List of reserved registers:

- \box255 is used for the contents of a page
- \count0-\count9 are used for page numbering

Scratch registers (freely available):

- \box0-\box254
- \count255
- \dimen0-\dimen9
- \muskip0-\muskip9
- \skip0-\skip9

Assign register using the '=' control character. For box registers, use the `\setbox` command instead.

```
\count255=17
\setbox\mybox=\hbox{blah}
```

You may use one of the following reservation macro to prevent any clash:

```
\newbox
\newcount
\newdimen
\newmuskip
```

```
\newskip
\newtoks
```

These macros use the following syntax: `\new*<csname>`. Example:

```
\newbox\mybox
\setbox\mybox=\hbox{blah}
```

These commands can not be used inside macros, otherwise every call to the macro would reserve another register.
You can print a register using the `\the` command. For counters use the `\number` command instead. For boxes use the `\box` command.

```
\the\hsize
\number\count255
\box\mybox
```

52.5. Arithmetic

The arithmetic capabilities of TeX are very limited, although this base suffice to extend it to some interesting features. The three main functions:

```
\advance <register> by <number>
\multiply <register> by <number>
\divide <register> by <number>
```

`register` may be of type count, dimen, muskip or skip. It does not make sense for box nor toks.

52.6. Conditionals

The base syntax is

```
\if* <test><true action>\fi
\if* <test><true action>\else<false action>\fi
```

where `\if*` is one command among the following.

Control sequence	**Description**
`\if <a><b>`	True if two character codes are equal.
`\ifcat <a><b>`	True if two category codes are equal.
`\ifdim <a><rel><b>`	Dimension relation, either < , > or = .
`\ifeof`	True if End-Of-File or non-existent file.
`\iffalse`	Always false.
`\ifhbox <reg>`	True if box register contains a horizontal box.
`\ifhmode`	True if in horizontal mode.
`\ifinner`	True if in internal mode.
`\ifmmode`	True if in math mode.
`\ifnum <a><rel><b>`	Number relation, either < , > or = .
`\ifodd <num>`	True if number is odd.
`\iftrue`	Always true.
`\ifvbox <reg>`	True if box register contains a vertical box.
`\ifvmode`	True if in vertical mode.
`\ifvoid <reg>`	True if box register is empty.
`\ifx <a><b>`	True if two macros expands to the same, or if two character codes are equal, or if two category codes are equal.

Example:

```
\ifnum 5>6
This is true
\else
This is false
\fi
```

This is false

52.6.1. Self defined conditionals

You can create new conditionals (as a kind of *boolean variables*) with the `\newif` command. With this self defined conditionals you can control the output of your code in an elegant way. The best way to illustrate the use of conditionals is through an example. Two versions of a document must be generated. One version for group A the other one for the rest of people (i.e. not belonging to group A):
1. We use `\newif` to define our conditional (i.e. boolean variable).

```
\newif\ifgroupA
```

2. In the following way we set a value (true or false) for our conditional

```
\groupAtrue % or
\groupAfalse
```

that is:

```
\<conditionalsname>true
\<conditionalsname>false
```

depending on which value we want to set in our conditional.
3. Now we can use our conditional anywhere after in an *if control structure* .

```
\ifgroupA
  % Here we write the code of the document that is
  % intended for the group A
\else
  % Here we write the code of the document that is
  % intended for the rest of the people
\fi
```

A full example is:

```
\newif\ifdirector

%I set the conditional to false
\directorfalse

\ifdirector
 I write something for the director.
\else
 I write something for common people.
\fi
```

I write something for common people.

52.6.2. Case statement

The syntax is `\ifcase <number><case0>\or<case1>\or...\else<defaultcase>\fi`. If `number` is equal to the case number, its content will be printed. Note that it starts at 0.

```
\ifcase 2 a\or b\or c\or d\else e\fi
```

c

`\else` is used to specify the default case (whenever none of the previous cases have matched).

52.7. Loops

The base syntax is

```
\loop <content> \if*<condition><true action>\repeat
```

As always, `content` and `true action` are arbitrary TeX contents. `\if*` refers to any of the conditionals[7]. Note that there is no `false action`, you cannot put an `\else` between `\if*` and `\repeat`. In some case this will be the opposite of what you want; you have to change the condition or to define a new conditional using `\newif`. Example:

```
\count255 = 1
\loop
  \TeX
\ifnum\count255 < 10
\advance\count255 by 1
\repeat
```

The above code will print TeX ten times.

52.8. Doing nothing

Sometimes it may be useful to tell TeX that you want to do nothing. There is two commands for that: `\relax` and `\empty`.
Classic example:

```
\def\myspace{\hskip 25pt\relax}
\myspace{} plus 10pt
```

The `\relax` prevents undesired behaviour if a `plus` or a `minus` is encounter after the command.
The difference between `\empty` and `\relax` lies in the expansion: `\empty` disappears after macro expansion.

7 Chapter 52.6 on page 586

52.9. TeX characters

52.9.1. *char*

We can print all characters using the `\char {charcode}` command. The charcode is actually the byte value. For example

```
\char65 = \char `A = \char `\A
```

Most characters correspond to the ASCII value (e.g. A-Za-z), some replace the non-printable characters from ASCII.

52.9.2. *chardef* and *mathchardef*

You can define control sequence to expand to a specific char. The syntax is `\chardef<control sequence>=<charcode>`. The following sequences do the same thing.

```
\chardef\myA=65
\chardef\myA=`A
\chardef\myA=`\A
```

Example:

```
\mathchardef\alphachar = "010B
$\alphachar$
```

52.9.3. Font encoding map

We can use the above primitive to print the font encoding map.

```
\count255 = 0
\loop
  [\number\count255 =\char\number\count255]
\ifnum\count255 < 127
\advance\count255 by 1
\repeat
```

Another version, with different fonts, one entry per line:

```
\count255 = 0
\loop
  [\number\count255 =
    \char\number\count255 \
    {\tt \char\number\count255}
    {\it \char\number\count255}
  ]
  \hfil\break
\ifnum\count255 < 127
\advance\count255 by 1
\repeat
```

52.10. Verbatim lines and spaces

It is rather confusing to discover (La)TeX treats all whitespace as the same type of spacing glue. Plain TeX provides some commands to preserve the spacing and newlines as you wrote it:

```
\begingroup
\obeylines
\obeyspaces
Relevant text here
\endgroup
```

which means that you will probably need to combine your own verbatim environment, and your command:

```
\newenvironment{myverbatim}{\begingroup \obeylines \obeyspaces}{\endgroup}
\newcommand{\mycommand}[n]{do something with #1 .. #n}
```

and then in your tex file:

```
\begin{myverbatim}
\mycommand{
whichever text it is important you
preserve the spacing and newslines
for, like when you want to generate
a verbatim block later on.
}
\end{myverbatim}
```

52.11. Macros defining macros

This is useful in some case, for example to define language commands as explained in Multilingual versions[8], where the end user can write

```
\en{some english text}
\de{some german text}
```

and make sure it switches to the appropriate Babel language.
Let's define a macros that will define language commands for instance. These commands are simple: if the argument is the value of the `\locale` variable, then the corresponding macro prints its content directly. Otherwise, it does nothing.
Basicly, what we want to do is extremely simple: define a bunch of macros like this:

```
\newcommand{\de}[1]{#1}
\newcommand{\en}[1]{}
\newcommand{\fr}[1]{}
```

In the previous snippet of code, only the `\de` command in going to output its content, `\en` and `\fr` will print nothing at all. That's what we want. The problem arises when you want to automate the task, or if you have a lot of languages, and you want to change the language selection. You just have to move the `#1`, but that's not convenient and it makes it impossible to choose the Babel language from command line. Think this out...
What we are going to do is to define the language commands dynamically following the value of the `\locale` variable (or any variable of your choice). Hence the use of the `\equal` command from the `ifthen` package.
Since it is hardly possible to write it in LaTeX, we will use some Plain TeX.

```
\def\locale{de}

\def\localedef#1{
  \ifthenelse{ \equal{\locale}{#1} }{
    %% Set the Babel language.
    %% Define the command to print the content.
  }{
    %% Define the command to print nothing.
  }
}
```

Another problem arises: how to define a command whose name is a variable? In most programming languages that's not possible at all. What we could be tempted to write is

```
\def\#1 #1{#1}
```

It will fail for two reasons.

8 Chapter 12.3 on page 135

1. The two last '#1' are supposed to refer to the arguments of the *new* macro, but they get expanded to the `\localedef` macro first argument because they are in the body of that macro.
2. `\#1` gets expanded to two tokens: '#' and '1', and the `\def` command will fail as it requires a valid control sequence name.

The solution to problem 1 is simple: use '##1', which will expand to '#1' when the macro is executed.

For problem 2, it is a little bit tricky. It is possible to tell `tex` that a specific token is a control sequence. This is what the `\csname...\endcsname` is used for. However

```
\def\csname#1\endcsname ##1{##1}
```

will fail because it will redefine `\csname` to '#1', which is not what we want, then `tex` will encounter `\endcsname`, which will result in an error.

We need to delay the expansion of `\def`, *i.e.* to tell `tex` to expand the `\csname` stuff first, then to apply `\def` on it. There is a command for that: `\expandafter{token1}{token2}`. It will expand `{token2}` before `{token1}`.

Finally if we want to set language from command line, we must be able to set the `\locale` variable so that the one in the source code is the default value that can be overridden by the one in the command line. This can be done with `\provdecommand`:

```
\providecommand\locale{fr}
```

The final code is

```
%% Required package.
\usepackage{ifthen}

%% TeX function that generates the language commands.
\def\localedef#1#2{
  \ifthenelse{ \equal{\locale}{#1} }{
    \selectlanguage{#2}
    \expandafter\def\csname#1\endcsname ##1{##1}
  }{
    \expandafter\def\csname#1\endcsname ##1{}
  }
}

%% Selected language. Can be placed anywhere before the language commands.
\providecommand\locale{fr}

%% Language commands.
\localedef{de}{ngerman}
\localedef{en}{english}
\localedef{fr}{frenchb}
%% ...
```

And you can compile with

```
latex '\providecommand\locale{en}\input{mydocument.tex}'
```

52.12. Notes and References

Further reading

- The TeXbook[9], *Donald Knuth*

9 `http://www.ctan.org/pkg/texbook`

- TeX by Topic[10], *Victor Eijkhout*
- TeX for the Impatient[11], *Paul W. Abrahams, Karl Berry and Kathryn A. Hargreaves*

10 `http://www.ctan.org/pkg/texbytopic`

11 `http://www.ctan.org/pkg/impatient`

53. Creating Packages

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 or class containing all your command and environment definitions. It can be made dynamic enough to fit to all your future documents.
Classes are `.cls` files, packages are stored in `.sty` files. They are very similar, the main difference being that you can load only one class per document.

53.1. `makeatletter` and `makeatother`

By default, LaTeX will allow the use of the '@' characters for control sequences from within package and class files, but not from within an end-user document. This way it is possible to *protect* commands, *i.e.* to make them accessible from packages only.
However it is possible to override this security with the duo `\makeatletter` and `\makeatother`. These commands only make sense in a regular document, they are not needed in package or class files.

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

\begin{document}

\makeatletter
\@author
\makeatother

\end{document}
```

53.2. Creating your own package

Your package can be made available in your document just like any other package: using the `\usepackage` command. Writing a package basically consists of copying the contents of your document preamble into a separate file with a name ending in `.sty` .
Let's write a first `custom.sty` file as an example package:

```
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesPackage{custom}[2013/01/13 Custom Package]

\RequirePackage{lmodern}

%% 'sans serif' option
\DeclareOption{sans}{
  \renewcommand{\familydefault}{\sfdefault}
}

%% 'roman' option
\DeclareOption{roman}{
  \renewcommand{\familydefault}{\rmdefault}
}
```

```
%% Global indentation option
\newif\if@neverindent\@neverindentfalse
\DeclareOption{neverindent}{
  \@neverindenttrue
}

\ExecuteOptions{roman}

\ProcessOptions\relax

%% Traditional LaTeX or TeX follows...
% ...

\newlength{\pardefault}
\setlength{\pardefault}{\parindent}
\newcommand{\neverindent}{ \setlength{\parindent}{0pt} }
\newcommand{\autoindent}{ \setlength{\parindent}{\pardefault} }

\if@neverindent
\neverindent
\fi

% ...

\endinput
```

- `\NeedsTeXFormat{...}` specifies which version of TeX or LaTeX is required at least to run your package. The optional date may be used to specify the version more precisely.
- `\ProvidesPackage` has to have the same name of the file without the extension. It 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. The date *is important* since it can be used by other package to specify the minimum version requirement they need to use your package.
- Next you may write some TeX or LaTeX code like loading package, but write only the bare minimum needed for the package options set below.
- `\RequirePackage` is equivalent to `\usepackage`.
- `\DeclareOptions` are end-user parameters. Each option is declared by one such command.
- `\ExecuteOptions{...}` tells which are the default.
- `\ProcessOptions\relax` terminates the option processing.
- Write whatever you want in it using all the LaTeX commands you know. Normally you should define new commands or import other packages.
- `\endinput`: this *must* be the last command.

Once your package is ready, we can use it in any document. Import your new package with the known command `\usepackage{mypack}`. The file `custom.sty` and the LaTeX source you are compiling must be in the same directory.

```
\documentclass{...}
\usepackage[neverindent,sans]{custom}
%...

\begin{document}

Blah...

\end{document}
```

For a more convenient use, it is possible to place the package within `$TEXMFHOME` (which is `~/texmf` by default) according to the TeX Directory Structure (TDS). That would be

```
$TEXMFHOME/tex/latex/custom/custom.sty
```

On Windows '~' is often `C:\Users\username` .
You may have to run `texhash` (or equivalent) to make your TeX distribution index the new file, thus making it available for use for any document. It will allow you to use your package as detailed above, but without it needing to be in the same directory as your document.

53.3. Creating your own class

It is also possible to create your own class file. The process is similar to the creation of your own package, you can call your own style file in the preamble of any document by the command:

```
\documentclass{myclass}
```

The name of the class file is then `myclass.cls` . Let's write a simple example:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{myclass}[2011/12/23 My Class]

%% Article options
\DeclareOption{10pt}{
  \PassOptionsToClass{\CurrentOption}{article}
}

%% Custom package options
\DeclareOption{sansserif}{
  \PassOptionsToPackage{\CurrentOption}{paxcommands}
}
\DeclareOption{neverindent}{
  \PassOptionsToPackage{\CurrentOption}{paxcommands}
}

%% Fallback
\DeclareOption*{
  \ClassWarning{myclass}{Unknown option '\CurrentOption'}
}

%% Execute default options
\ExecuteOptions{10pt}

%% Process given options
\ProcessOptions\relax

%% Load base
\LoadClass[a4paper]{article}

%% Load additional packages and commands.
\RequirePackage{custom}

%% Additional TeX/LaTeX code...

\endinput
```

- `\ProvidesClass` is the counterpart of `\ProvidesPackage`.
- `\PassOptionsToClass` and `\PassOptionsToPackage` are used to automatically invoke the corresponding options when the class or the package is loaded.

- `\DeclareOption*`: the starred version lets you handle non-implemented options.
- `\ClassWarning` will show the corresponding message in the TeX compiler output.
- `\LoadClass` specifies the unique parent class, if any.

53.4. Hooks

There are also hooks for classes and packages.
- `\AtEndOfPackage`
- `\AtEndOfClass`

They behave as the document hooks. See LaTeX Hooks[1].

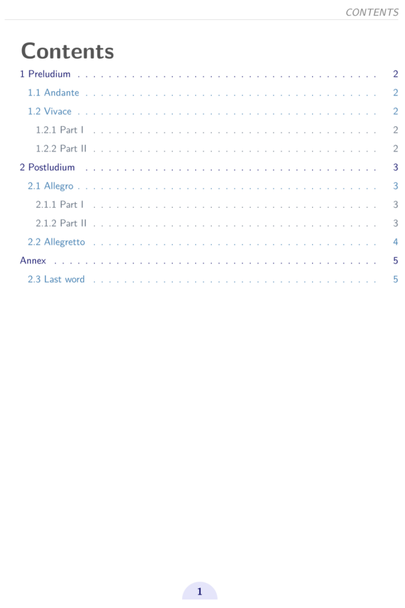
1 Chapter 51.9 on page 576

54. Themes

Newcomers to LaTeX often feel disappointed by the lack of visual customization offered by the system. Actually this is done on purpose: the LaTeX philosophy takes a point at doing the formatting while the writer focuses on the content.
In this chapter, we will show what we can achieve with some efforts.

54.1. Introduction

In the following we will write the theme, a package that will only change the appearance of the document, so that our document will work with or without the theme.
Note that if it may look eye-candy, this is absolutely not a model of typography. You should not use such theme for serious publications. This is more a technogical example to exhibit LaTeX capabilities.



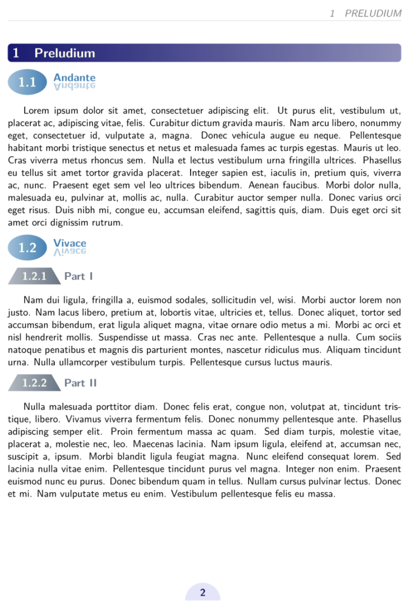
CONTENTS

Contents

1 Preludium 2
1.1 Andante 2
1.2 Vivace 2
1.2.1 Part I 2
1.2.2 Part II 2
2 Postludium 3
2.1 Allegro 3
2.1.1 Part I 3
2.1.2 Part II 3
2.2 Allegretto 4
Annex 5
2.3 Last word 5

1

Figure 197 Custom theme (TOC)



1 PRELUDIUM

1 Preludium

1.1 Andante

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1.2 Vivace

1.2.1 Part I

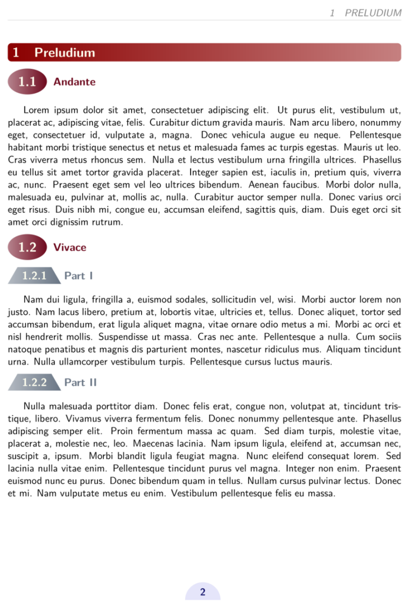
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1.2.2 Part II

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2

Figure 198 Custom theme



1 PRELUDIUM

1 Preludium

1.1 Andante

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1.2 Vivace

1.2.1 Part I

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1.2.2 Part II

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2

Figure 199 Custom theme (red)

54.2. Package configuration

Nothing much to say here. This is a direct application of the Creating Packages[1] chapter. We load the required packages.
• `needspace` is used to prevent page break right after a sectioning command.
• `tikz` is used to draw the fancy material.
We define a color option, you may use as much as you want. Defining colors with specific names makes it very flexible. We also use an option to toggle the fancy reflection effect which might be a little *too much* !

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{theme-fancy}[2013/01/13 v1.0 fancy theme]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Packages

\RequirePackage{geometry}
\RequirePackage{needspace}
\RequirePackage[svgnames]{xcolor}

\RequirePackage{hyperref}
\hypersetup{colorlinks=true}

\RequirePackage{fancyhdr}

\RequirePackage{tikz}
\usetikzlibrary{
  calc,
  decorations.pathmorphing,
  fadings,
  shadows,
  shapes.geometric,
  shapes.misc,
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Options

%% Toggle reflection.
\newif\if@mirrors\@mirrorsfalse
\DeclareOption{mirrors}{
  \@mirrorstrue
}

%% Colors.
\newif\if@red\@redfalse
\DeclareOption{red}{
  \@redtrue
}

\ExecuteOptions{}
\ProcessOptions\relax

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Configuration

\renewcommand{\familydefault}{\sfdefault}
\setlength{\parskip}{0.5\baselineskip}

%% Colors
\colorlet{toctitle}{DarkGray!50!black}
\colorlet{titlebg}{MidnightBlue}
```

1 Chapter 53 on page 593

```
\colorlet{titlefg}{LightBlue}
\colorlet{titletxt}{MidnightBlue}
\colorlet{sectionfg}{MidnightBlue}
\colorlet{subsectionfg}{SteelBlue}
\colorlet{subsubsectionfg}{LightSteelBlue!60!black}

\if@red
\colorlet{toctitle}{DarkGray!50!black}
\colorlet{titlebg}{DarkRed}
\colorlet{titlefg}{FireBrick!50}
\colorlet{titletxt}{DarkRed}
\colorlet{sectionfg}{DarkRed}
\colorlet{subsectionfg}{Crimson!50!black}
\colorlet{subsubsectionfg}{LightSteelBlue!60!black}
\fi
```

54.3. Header and footer

We use TikZ to draw a filled semicircle.
`fancyhdr` is used to set header and footer. We take care of using the `fancy` style and to start from scratch by erasing the previous header and footer with `\fancyhf{}`.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Header and Footer

\tikzstyle{foliostyle}=[fill=Lavender, text=MidnightBlue, inner sep=5pt,
 semicircle]

\pagestyle{fancy}
\fancyhf{}
\fancyfoot[C]{
  \vskip 3pt
  \begin{tikzpicture}
    \node[foliostyle]
    {\bfseries\thepage};
  \end{tikzpicture}
}

\renewcommand{\headrulewidth}{0.8pt}
\addtolength{\headheight}{\baselineskip}
\renewcommand{\headrule}{\color{LightGray}\hrule}
\fancyhead[LE]{ \textcolor{gray}{\slshape \rightmark} }
\fancyhead[RO]{ \textcolor{gray}{\slshape \leftmark} }
```

54.4. Table of contents

We redefine commands used by `\tableofcontents`.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Table of contents

\newcommand{\sectiontoccolor}{sectionfg}
\newcommand{\subsectiontoccolor}{subsectionfg}
\newcommand{\subsubsectiontoccolor}{subsubsectionfg}

\renewcommand*\l@section{\color{\sec
tiontoccolor}\def\@linkcolor{\sectiontoccolor}\@dottedtocline{1}{1.5em}{2.3em}}
\renewcommand*\l@subsection{\color{\subsectio
ntoccolor}\def\@linkcolor{\subsectiontoccolor}\@dottedtocline{1}{2.3em}{3.1em}}
\renewcommand*\l@subsubsection{\color{\subsubsectionto
ccolor}\def\@linkcolor{\subsubsectiontoccolor}\@dottedtocline{1}{3.1em}{3.9em}}
\def\contentsline#1#2#3#4{%
  \ifx\\#4\\%
  \csname l@#1\endcsname{#2}{#3}%
  \else
  \csname l@#1\endcsname{\hyper@linkstart{link}{#4}{#2}\hyper@linkend}{%
    \hyper@linkstart{link}{#4}{#3}\hyper@linkend
  }%
  \fi
}

%% New title format -- 'section' is used by default.
\newcommand{\tocformat}[1]{{\Huge\bf#1}}

\renewcommand\tableofcontents{%
  \tocformat{
    \textcolor{toctitle}{\contentsname}
    \@mkboth{\MakeUppercase\contentsname}{\MakeUppercase\contentsname}
  }%
  \@starttoc{toc}%
}
```

54.5. Sectioning

This is definitely the most complex part. It is not that hard since the code is almost the same for `\section`, `\subsection` and `\subsubsection`.
We use `\needspace` to make sure there is no line break right after a sectioning command. We enclose the command in a group where we set a font size since the space we need is `\baselineskip` which depends on the font size.
Starred commands will not set the counters (LaTeX detault behaviour). You can choose to handle starred command differently by resetting the counters for instance.
We preceed the section printing by a `\noindent`. We make sure to end the section printing by a `\par` command to make sure following text gets printed properly.
For `\subsection` we make use of the `mirrors` option to change the appearance accordingly.
To handle the PDF bookmarks properly we need the following lines at the end of the definitions.

```
\phantomsection
\addcontentsline{toc}{section}{\thesection~#1}
```

Finally, for `\section` only, we want it to print in the header, so we call the `\sectionmark` command. Here we changed the behaviour of the starred command over the original LaTeX version, since we define and use the `\sectionmarkstar` command.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Section style

\renewcommand\section{
  \@ifstar
  \my@sectionstar
  \my@section
}

%% Note: to justify, text width must be set to \textwidth - 2*(inner sep).
\tikzstyle{sectionstyle}=[
  inner sep=5pt,
  text width=\textwidth-10pt,
  left color=sectionfg!100!white,
  right color=sectionfg!50!white,
  rounded corners,
  text=Ivory,
  rectangle
]
```

```
\newcommand\my@section[1]{
  \stepcounter{section}
  {\Large\needspace{\baselineskip}}
  \noindent
  \begin{tikzpicture}
    \node[sectionstyle] {\bfseries\Large\thesection\quad#1};
  \end{tikzpicture}
  \par
  \phantomsection
  \addcontentsline{toc}{section}{\thesection~#1}
  \sectionmark{#1}
}

\newcommand{\sectionmarkstar}[1]{\markboth{\MakeUppercase{#1}}{}}

\newcommand\my@sectionstar[1]{
  {\Large\needspace{\baselineskip}}
  \noindent
  \begin{tikzpicture}
    \node[sectionstyle] {\bfseries\Large#1};
  \end{tikzpicture}
  \par
  \phantomsection
  \addcontentsline{toc}{section}{#1}
  \sectionmarkstar{#1}
}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Subsection style

\renewcommand\subsection{
  \@ifstar
  \my@subsectionstar
  \my@subsection
}

\tikzstyle{subsectionstyle}=[
  left color=subsectionfg!50!white,
  right color=subsectionfg!100!white,
  text=Ivory,
  ellipse,
  inner sep=5pt
]

\newcommand\my@subsection[1]{
  \stepcounter{subsection}
  {\Large\needspace{\baselineskip}}
  \noindent
  \begin{tikzpicture}
    \node[subsectionstyle,anchor=west] (number) at (0,0)
 {\bfseries\Large\thesubsection};
    \if@mirrors
    \node[above right,subsectionfg,anchor=south west] at
 ($(number.east)+(0.1,-0.1)$) {\large\bfseries#1};
    \node[yscale=-1, scope fading=south, opacity=0.4, above, anchor=south west,
 subsectionfg] at ($(number.east)+(0.1,0.1)$) {\large\bfseries#1};
    \else
    \node[above right,subsectionfg,anchor=west] at ($(number.east)+(0.1,0)$)
 {\large\bfseries#1};
    \fi
  \end{tikzpicture}
  \par
  \phantomsection
  \addcontentsline{toc}{subsection}{\thesubsection~#1}
}
```

```
\newcommand\my@subsectionstar[1]{
  {\Large\needspace{\baselineskip}}
  \noindent
  \begin{tikzpicture}
    \node[subsectionstyle,anchor=west] (number) at (0,0)
 {\bfseries\Large\phantom{1}};
    %
    \if@mirrors
    \node[above right,subsectionfg,anchor=south west] at
 ($(number.east)+(0.1,-0.1)$) {\large\bfseries#1};
    \node[yscale=-1, scope fading=south, opacity=0.4, above, anchor=south west,
 subsectionfg] at ($(number.east)+(0.1,0.1)$) {\large\bfseries#1};
    \else
    \node[above right,subsectionfg,anchor=west] at ($(number.east)+(0.1,0)$)
 {\large\bfseries#1};
    \fi
  \end{tikzpicture}
  \par
  \phantomsection
  \addcontentsline{toc}{subsection}{#1}
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Subsubsection style

\renewcommand\subsubsection{
  \@ifstar
  \my@subsubsectionstar
  \my@subsubsection
}

\tikzstyle{subsubsectionstyle}=[
  left color=subsubsectionfg!50!white,
  right color=subsubsectionfg!100!white,
  text=Ivory,
  shape=trapezium,
  inner sep=5pt
]

\newcommand\my@subsubsection[1]{
  \stepcounter{subsubsection}
  \noindent
  \begin{tikzpicture}
        \node[subsubsectionstyle] (number) {\bfseries\large\thesubsubsection};
        \node[subsubsectionfg, right of=number, anchor=west] {\large\bfseries#1};
  \end{tikzpicture}
  \par
  \phantomsection
  \addcontentsline{toc}{subsubsection}{\thesubsubsection~#1}
}

\newcommand\my@subsubsectionstar[1]{
  \noindent
  \begin{tikzpicture}
        \node[subsubsectionstyle] (number) {\bfseries\large\vphantom{1}};
        \node[subsubsectionfg, right of=number, anchor=west] {\large\bfseries#1};
  \end{tikzpicture}
  \par
  \phantomsection
  \addcontentsline{toc}{subsubsection}{#1}
}

\endinput
```

54.6. Notes and References

Part IX.

Miscellaneous

55. Modular Documents

During this guide we have seen what is possible to do and how this can be achieved, but the question is: I want to write a proper text with LaTeX, what to do then? Where should I start from? This is a short step-by-step guide about how to start a document properly, keeping a good high-level structure. This is all about organizing your files using the modular capabilities of LaTeX. This way it will be very easy to make modifications even when the document is almost finished. These are all just suggestions, but you might take inspiration from that to create your own document.

55.1. Project structure

Create a clear structure of the whole project this way:

1. create a directory only for the project. We'll refer to that in the following parts as the *root directory*
2. create two other directories inside the root, one for LaTeX documents, the other one for images. Since you'll have to write their name quite often, choose short names. A suggestion would be simply *tex* and *img* .
3. create your document (we'll call it document.tex, but you can use the name you prefer) and your own package (for example *mystyle.sty*); this second file will help you to keep the code cleaner.

If you followed all those steps, these files should be in your root directory, using "/" for each directory:

```
./document.tex
./mystyle.sty
./tex/
./img/
```

nothing else.

55.2. Getting LaTeX to process multiple files

As your work grows, your LaTeX file can become unwieldy and confusing, especially if you are writing a long article with substantial, discrete sections, or a full-length book. In such cases it is good practice to split your work into several files. For example, if you are writing a book, it makes a lot of sense to write each chapter in a separate `.tex` file. LaTeX makes this very easy thanks to two commands:

```
\input{filename}
```

and

```
\include{filename}
```

55.2.1. Comparing the methods: input vs include

The differences between these two ways to include files is explained below. What they have in common is that they process the contents of `filename.tex` before continuing with the rest of the base file (the file that contains these statements). When the compiler processes your base file and reaches one of the commands `\input` or `\include`, it reads `filename.tex` and processes its content in accordance with the formatting commands specified in the base file. This way you can put all the formatting options in your base file and write the contents using `\input` or `\include` in the files which contain the actual content of your work. This means that the important part of your working process, i.e. writing, is kept largely separate from formatting choices. This is one of the main reasons why LaTeX is so good for serious writing! You will thus be dealing solely with text and very basic commands such as `\section`, `\emph` etc. Your document will be uncluttered and much easier to work with.
The second method of including a file, `\include{filename}`, differs from the first in some important ways. You cannot nest `\include` statements within a file added via `\include`, whereas `\input`, on the other hand, allows you to call files which themselves call other files, ad infinitum (well, nearly!). You can, however, `\include` a file which contains one or more `\input` commands. Please resist the temptation to nest files in this way simply because the system can do it: you will end up with just another kind of complexity!
Another important difference is that using `\include` will force a page break (which makes it ideal for a book's chapters), whereas the `\input` command does not (which in turn makes it ideal for, say, a long article with discrete sections, which of course are not normally set on a new page).
In either case the `.tex` filename extension is optional.
Working on discrete parts of your documents has consequences for how the base file is compiled; these will be dealt with below.

55.2.2. Using different paths

When the LaTeX compiler finds a reference to an external file in the base file, it will look for it in the same directory. However, you can in principle refer to any file on your system, using both absolute and relative paths.
An *absolute* path is a full path- and filename with every element specified. So, `filename.tex` might have the full path,

```
\input{/home/user/texfiles/filename.tex}
```

If you had created the directory `myfiles` for your writing project, in your `texfiles` directory, its full path would be,

```
\input{/home/user/texfiles/myfiles/filename.tex}
```

Obviously, using absolute paths is inefficient if you are referring to a file in the current directory. If, however, you need to include a file which is always kept at a specific place in your system, you may refer to it with an absolute path, for example,

```
\input{/home/user/documents/useful/foo.tex}
```

In practice, an absolute file path is generally used when one has to refer to a file which is quite some way away in the file system (or perhaps even on a different server!). One word of warning: do not leave empty spaces in the filenames, they can cause ambiguous behaviour. Either leave no spaces or use underscores _ instead.

You may, however, need to make your source portable (to another computer or to a different location of your harddisk), in which case *relative* paths should be used if you wish to avoid unnecessary rewriting of path names. Or, a relative path may simply be a more efficient and elegant way of referring to a file. A relative path is one which is defined in relation to the current directory, in our case the one which contains the base file. LaTeX uses the standard UNIX notation: with a simple dot **.** you refer to the current directory, and by two dots **..** you refer to the previous directory, that is the one above the current directory in the file system tree. The slash / is used to separate the different components of a pathname: directories and filenames. So by **./** you refer to the current directory, by **../** you refer to the previous directory, by **../../** you refer to a directory which is two steps upwards in the filesystem tree. Writing

```
\input{./filename.tex}
```

will have *exactly* the same effect as writing

```
\input{filename.tex}
```

but if you found it more convenient to put all your files in a sub-directory of your current directory, called `myfiles` , you would refer to that file by specifying

```
\input{./myfiles/filename.tex}
```

Indeed, in our example of the absolute path above, you could refer to that file relatively, too:

```
\input{../../documents/useful/foo.tex}
```

Of course, all commonly used file systems – Linux, Mac OS X and Windows – also feature the UNIX **./** , **../** facility outlined above. Do note, however, that LaTeX uses forward slashes / even on Microsoft Windows platforms, which use backslashes \ in pathnames. LaTeX implementations for Windows systems perform this conversion for you, which ensures that your document will be valid across all installations.
This flexibility, inherent in the way in which LaTeX is integrated with modern file systems, lets you input files in a way which suits your particular set-up.
When using relative paths within a LaTeX file imported by \input or \include, it is important to note that the paths are relative to the directory in which the main .tex file resides, not to the directory in which the included (or input) file is found. This is likely to be an issue if using a folder per chapter, with the figures in each chapter's folder, and using \include to read the chapter source into a main LaTeX file in a parent folder.

55.2.3. Compiling the base file

When you compile your document, page references and the like will change according to your use of the `\input` and `\include` commands. Normally LaTeX users only run the compiler on parts of the document to check that an individual chapter is syntactically correct and looks as the writer intended. A full run is generally only performed for producing a full draft or the final version. In such cases, it is invariably necessary to run LaTeX twice or more to resolve all the page numbers, references, etc. (especially if you are using bibliographic software such as BiBTeX, too).
The simplest way to check that one or more of the various components of your work is syntactically robust, is to comment out the command with a percentage sign, for example:

```
\documentclass{article}
\begin{document}
%\input{Section_1}
```

```
%\input{Section_2}
%\input{Section_3}
\input{Section_4}
%\input{Section_5}
\end{document}
```

This code will process your base file with the `article` conventions but only the material in the file `Section_4.tex` will be processed. If that was, say, the last thing you needed to check before sending off to that major journal, you would then simply remove all the percentage signs and rerun LaTeX, repeating the compiling process as necessary to resolve all references, page numbers and so on.

55.2.4. Using \includeonly

Using this command provides more complex, and hence more useful possibilities. If you include the following command in your preamble, i.e. before \begin{document},

```
\includeonly{filename1,filename2,...}
```

only the files specified between the curly braces will be included. Note that you can have one or more files as the argument to this command: separate them with a comma, no spaces.
This requires that there are `\include` commands in the document which specify these files. The filename should be written without the `.tex` file extension:

```
\documentclass{book}
\includeonly{Chapter_1,Chapter_4}  % compile just chapters 1 and 4, space
 characters not permitted
\begin{document}
\include{Chapter_1}                % omit the '.tex' extension
\include{Chapter_2}
\include{Chapter_3}
\include{Chapter_4}
\end{document}
```

This code would process the base file but only include the content of the author's first and fourth chapters (`Chapter_1.tex` and `Chapter_4.tex`). Importantly, this alternative retains as much of the `.aux` information as possible from the previous run, so messes up your cross-references much less than the makeshift suggestion above.

55.2.5. Separate compilation of child documents

A disadvantage of solely using `\input` and `\include` is that only the base document can be compiled. However, you may decide that you work better on individual sections of text and wish to edit and compile those separate from the main file. There are a few packages available to address this problem.

Subfiles

The subfiles package[1] provides a way to compile sections of a document using the same preamble as the main document.
In the main document, the package must be loaded as:

```
\usepackage{subfiles}
```

Instead of using `\input` and `\include` , child documents must be loaded as follows:

1 `http://www.ctan.org/pkg/subfiles`

```
\subfile{filename}
```

The child documents must start with the following statements:

```
\documentclass[main.tex]{subfiles}
\begin{document}
```

and end with:

```
\end{document}
```

In summary, the base document (`main.tex`) looks like:

```
\documentclass{book}
\usepackage{subfiles}
\begin{document}
%% my document content
\subfile{chapter1}
%% more of my document content
\end{document}
```

and Chapter 1 (`chapter1.tex`) looks like:

```
\documentclass[main.tex]{subfiles}
\begin{document}
%% my chapter 1 content
%%
%% more of my chapter 1 content
\end{document}
```

Some linux distributions don't have subfiles package in their latex distributions, since it was not included until TeXLive 2012. You can download subfiles.tds.zip[2] from CTAN. This package will contain two files `subfiles.cls` and `subfiles.sty` . Move these files to a directory under the name `subfiles` in the path `/usr/share/texmf/tex/latex` . This still won't make the package available; the `texhash` program must be executed first. Now you are good to go!

Standalone

The standalone package[3] is designed for moving more of the opposite direction than subfiles. It provides a means for importing the preamble of child documents into the main document, allowing for a flexible way to include text or images in multiple documents (e.g. an article and a presentation[4]).
In the main document, the package must be loaded as:

```
\usepackage{standalone}
```

Child documents are loaded using `\input` or `\include` .
The child documents contain, for example, the following statements:

```
\documentclass{standalone}
% Load any packages needed for this document
\begin{document}
% Your document or picture
\end{document}
```

In summary, the base document (`main.tex`) looks like:

```
\documentclass{book}
\usepackage{standalone}
```

2 http://mirrors.ctan.org/install/macros/latex/contrib/subfiles.tds.zip
3 http://www.ctan.org/pkg/standalone
4 Chapter 41 on page 491

```
\begin{document}
%% my document content
\input{chapter1}
%% more of my document content
\end{document}
```

and Chapter 1 (`chapter1.tex`) looks like:

```
\documentclass{standalone}
% Preamble
\begin{document}
%% my chapter 1 content
%%
%% more of my chapter 1 content
\end{document}
```

55.2.6. Inserting PDF files

If you need to insert an existing, possibly multi-page, PDF file into your LaTeX document, whether or not the included PDF was compiled with LaTeX or another tool, consider using the pdfpages package[5]. In the preamble, include the package:

```
\usepackage[final]{pdfpages}
```

This package also allows you to specify which pages you wish to include: for example, to insert pages 3 to 6 from some file `insertme.pdf` , use:

```
\includepdf[pages=3-6]{insertme.pdf}
```

To insert the whole of `insertme.pdf` :

```
\includepdf[pages=-]{insertme.pdf}
```

For full functionality, compile the output with `pdflatex` .
Additional information can be found in the chapter Export To Other Formats[6].

55.3. The file `mystyle.sty`

Instead of putting all the packages you need at the beginning of your document as you could, the best way is to load all the packages you need inside another dummy package called *mystyle* you will create just for your document. The good point of doing this is that you will just have to add one single `\usepackage` in your document, keeping your code much cleaner. Moreover, all the info about your style will be within one file, so when you will start another document you'll just have to copy that file and include it properly, so you'll have exactly the same style you have used.
Creating your own style is very simple: create a file called `mystyle.sty` (you could name it as you wish, but it has to end with ".sty"). Write at the beginning:

```
\ProvidesPackage{mystyle}
```

Then add all the packages you want with the standard command `\usepackage{...}` as you would do normally, change the value of all the variables you want, etc. It will work like the code you put here would be copied and pasted within your document.
While writing, whenever you have to take a decision about formatting, define your own command for it and add it to your `mystyle.sty` :let LaTeX work for you. If you do so, it will be very easy to change it if you change your mind.

5 `http://www.ctan.org/tex-archive/macros/latex/contrib/pdfpages/`
6 Chapter 57.3.3 on page 629

This is actually the beginning of the process of writing a package. See LaTeX/Macros[7] for more details.
For a list of several packages you can use, see the List of Packages[8] section.

55.4. The main document `document.tex`

Then create a file called `document.tex` ; this will be the main file, the one you will compile, even if you shouldn't need to edit it very often because you will be working on other files. It should look like this (it's the sample code for a *report* , but you might easily change it to *article* or whatever else):

```
\documentclass[12pt,a4paper]{report}
\usepackage{graphicx}
\usepackage{ifpdf}
\ifpdf
   % put here packages only for the PDF:
   \DeclareGraphicsExtensions{.pdf,.png,.jpg,.mps}
   \usepackage{hyperref}
\else
   % put here packages only for the DVI:
\fi

% put all the other packages here:

\usepackage{mystyle}

\begin{document}

\input{./tex/title.tex}
%\maketitle
\tableofcontents
\listoffigures
\listoftables

\input{./tex/intro.tex}
\input{./tex/main_part.tex}
\input{./tex/conclusions.tex}

\appendix
\input{./tex/myappendix.tex}


% Bibliography:
\clearpage
\addcontentsline{toc}{chapter}{Bibliography}
\input{./tex/mybibliography.tex}

\end{document}
```

Here a lot of code expressed in previous sections has been used. At the beginning there is the header discussed in the Tips & Tricks[9] section, so you will be able to compile in both DVI and PDF. Then you import the only package you need, that is your *mystyle.sty* (note that in the code it has to be imported without the extension), then your document starts. Then it inserts the title: we don't like the output of `\maketitle` so we created our own, the code for it will be in a file called `title.tex` in the folder called `tex` we

7 Chapter 51 on page 569
8 `http://en.wikibooks.org/wiki/LaTeX%2FPackages%23Packages_list`
9 Chapter 59 on page 643

created before. How to write it is explained in the Title Creation[10] section. Then tables of contents, figure and tables are inserted. If you don't want them, just comment out those lines. Then the main part of the document in inserted. As you can see, there is no text in `document.tex` : everything is in other files in the `tex` directory so that you can easily edit them. We are separating our text from the structural code, so we are improving the "What You See is What You Mean" nature of LaTeX. Then we can see the appendix and finally the Bibliography. It is in a separate file and it is manually added to the table of contents using a tip suggested in the Tips & Tricks[11].
Once you have created your `document.tex` you won't need to edit it anymore, unless you want to add other files in the `tex` directory, but this is not going to happen very often. Now you can write your document, separating it into as many files as you want and adding many pictures without getting confused: thanks to the rigid structure you gave to the project, you will be able to keep track of all your edits clearly.
A suggestion: do not give your files names like "chapter_01.tex" or "figure_03.png", i.e. try to avoid using numbers in file-names: if the numbering LaTeX gives them automatically, is different from the one you gave (and this will likely happen) you will get really confused. When naming a file, stop for a second, think about a short name that can fully explain what is inside the file without being ambiguous, it will let you save a lot of time as soon as the document gets larger.

55.5. External Links

- Subfiles package documentation[12]
- Standalone package documentation[13]
- pdfpages package documentation[14]

10 Chapter 15 on page 187
11 Chapter 59 on page 643
12 `http://mirrors.ctan.org/macros/latex/contrib/subfiles/subfiles.pdf`
13 `http://mirrors.ctan.org/macros/latex/contrib/standalone/standalone.pdf`
14 `http://mirrors.ctan.org/macros/latex/contrib/pdfpages/pdfpages.pdf`

56. Collaborative Writing of LaTeX Documents

Note: This Wikibook is based on the article *Tools for Collaborative Writing of Scientific LaTeX Documents* [1] by Arne Henningsen[2] that is published in *The PracTeX Journal* 2007, number 3 (`http://www.tug.org/pracjourn/`).

56.1. Abstract

Collaborative writing of documents requires a strong synchronisation among authors. This Wikibook describes a possible way to organise the collaborative preparation of LaTeX documents. The presented solution is primarily based on the version control system *Subversion* (`http://subversion.apache.org/`). The Wikibook describes how *Subversion* can be used together with several other software tools and LaTeX packages to organise the collaborative preparation of LaTeX documents.

56.1.1. Other Methods

- You can use one of the online solutions listed in the Installation[3] chapter. Most of them have collaboration features.
- Another option for collaboration is dropbox[4]. It has 2 GB free storage and versioning system. Works like SVN, but more automated and therefore especially useful for beginning LaTeX users. However, Dropbox is not a true versioning control system, and as such it does not allow you to roll the article back to previous versions.
- You can use an online collaborative tool built on top of a versioning control system, such as Authorea[5] or ShareLatex[6]. Authorea performs most of the actions described in this document, but in the background (it is built on Git). It allows authors to enter LaTeX or Markdown via a GUI with mathematical notation, figures, d3.js plots, IPython notebooks, data, and tables. All content is rendered to HTML5. Authorea also features a commenting system and article-based chat to ease collaboration and review.
- As the LaTeX system uses plain text, you can use synchronous collaborative editors like Gobby[7]. In Gobby you can write your documents in collaboration with anyone in real time. It is strongly recommended that you use utf8 encoding (especially if there

1 `http://tug.org/pracjourn/2007-3/henningsen/`
2 `http://en.wikibooks.org/wiki/User%3AArnehe`
3 Chapter 2 on page 11
4 `http://www.getdropbox.com`
5 `https://authorea.com/`
6 `https://www.sharelatex.com/`
7 `http://en.wikipedia.org/wiki/Gobby`

are users on multiple operating systems collaborating) and a stable network (typically wired networks).

- TitanPad[8] (or other clones[9] of EtherPad[10]). To compile use the command:
  ```
  wget -O filename.tex "http://titanpad.com/ep/pad/export/xxxx/latest?format=txt"
  && (latex filename.tex)
  ```
  where 'xxxx' should be replaced by the pad number (something like 'z7rSrfrYcH').
- With a dedicated Linux box with LaTeX & Dropbox it's possible to use Google docs and some scripting[11] to get automatically generated PDFs on Dropbox from updates on Google Docs.
- You can use a distributed version control system[12] such as Mercurial[13] or Git[14]. This is the definitive solution for users looking for control and advanced features like branch and merge. The learning curve will be steeper than that for a web-based solution.

56.2. Introduction

The collaborative preparation of documents requires a considerable amount of coordination among the authors. This coordination can be organised in many different ways, where the best way depends on the specific circumstances.
In this Wikibook, I describe how the collaborative writing of LaTeX documents is organised at our department (Division of Agricultural Policy, Department of Agricultural Economics, University of Kiel, Germany). I present our software tools, and describe how we use them. Thus, this Wikibook provides some ideas and hints that will be useful for other LaTeX users who prepare documents together with their co-authors.

56.3. Interchanging Documents

There are many ways to interchange documents among authors. One possibility is to compose documents by interchanging e-mail messages. This method has the advantage that common users generally do not have to install and learn the usage of any extra software, because virtually all authors have an e-mail account. Furthermore, the author who has modified the document can easily attach the document and explain the changes by e-mail as well. Unfortunately, there is a problem when two or more authors are working at the same time on the same document. So, how can authors synchronise these files?
A second possibility is to provide the document on a common file server, which is available in most departments. The risk of overwriting each others' modifications can be eliminated by locking files that are currently edited. However, generally the file server can be only accessed from within a department. Hence, authors who are out of the building cannot use this method to update/commit their changes. In this case, they will have to use another way to overcome this problem. So, how can authors access these files?

8 http://titanpad.com
9 http://etherpad.org/etherpadsites.html
10 http://en.wikipedia.org/wiki/EtherPad
11 https://gist.github.com/1995648
12 http://en.wikipedia.org/wiki/Distributed_revision_control
13 http://en.wikipedia.org/wiki/Mercurial
14 http://en.wikibooks.org/wiki/Git

A third possibility is to use a version control system. A comprehensive list of version control systems can be found at Wikipedia[15]. Version control systems keep track of all changes in files in a project. If many authors modify a document at the same time, the version control system tries to merge all modifications automatically. However, if multiple authors have modified the same line, the modifications cannot be merged automatically, and the user has to resolve the conflict by deciding manually which of the changes should be kept. Authors can also comment their modifications so that the co-authors can easily understand the workflow of this file. As version control systems generally communicate over the internet (e.g. through TCP/IP connections), they can be used from different computers with internet connections. A restrictive firewall policy might prevent the version control system from connecting to the internet. In this case, the network administrator has to be asked to open the appropriate port. The internet is only used for synchronising the files. Hence, a permanent internet connection is not required. The only drawback of a version control system could be that it has to be installed and configured. Moreover, a version control system is useful even if a single user is working on a project. First, the user can track (and possibly revoke) all previous modifications. Second, this is a convenient way to have a backup of the files on other computers (e.g. on the version control server). Third, this allows the user to easily switch between different computers (e.g. office, laptop, home).

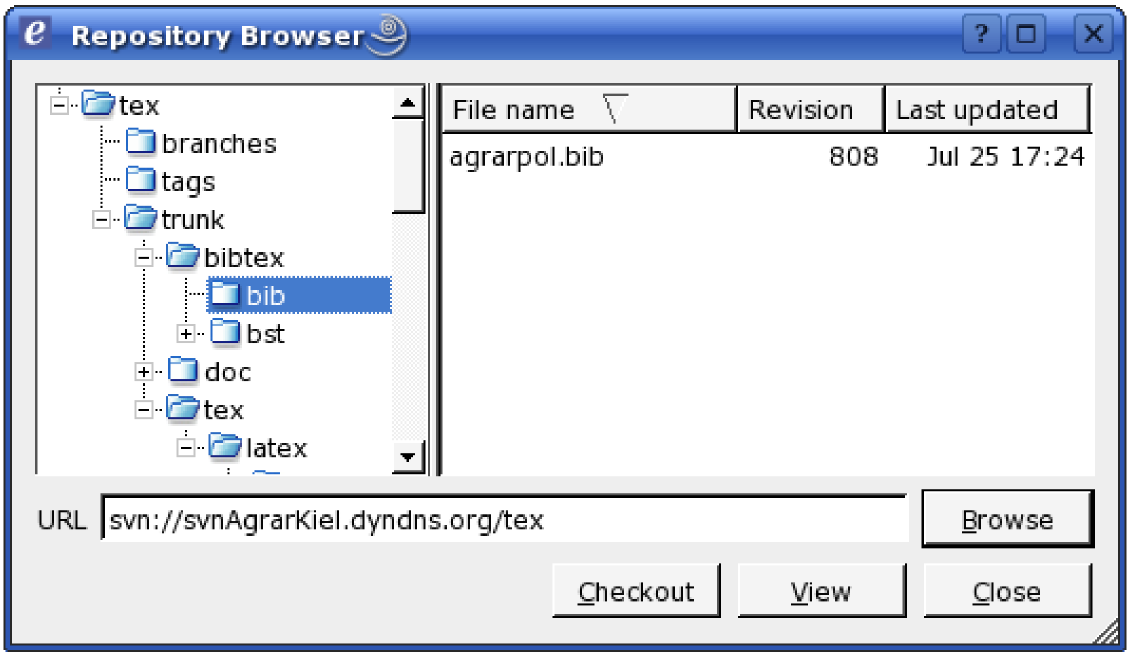
56.4. The Version Control System *Subversion*

Subversion (SVN)[16] comes as a successor to the popular version control system CVS. SVN operates on a client-server model in which a central server hosts a project repository that users copy and modify locally. A repository functions similarly to a library in that it permits users to check out the current project, make changes, and then check it back in. The server records all changes a user checks in (usually with a message summarizing what changes the user made) so that other users can easily apply those changes to their own local files.
Each user has a local *working copy* of a remote *repository* . For instance, users can *update* changes from the repository to their working copy, *commit* changes from their own working copy to the repository, or (re)view the differences between working copy and repository.
To set up a SVN version control system, the SVN server software has to be installed on a (single) computer with permanent internet access. (If this computer has no static IP address, one can use a service like DynDNS[17] to be able to access the server with a static hostname.) It can run on many Unix, modern MS Windows, and Mac OS X platforms.
Users do not have to install the SVN server software, but a SVN ”client” software. This is the unique way to access the repositories on the server. Besides the basic SVN command-line client, there are several Graphical User Interface Tools (GUIs) and plug-ins for accessing the SVN server (see `http://subversion.tigris.org/links.html`). Additionally, there are very good manuals about SVN freely available on the internet (e.g. `http://svnbook.red-bean.com`).

15 `http://en.wikipedia.org/wiki/List_of_revision_control_software`
16 `http://subversion.apache.org/`
17 `http://www.dyndns.com/`

At our department, we run the SVN server on a GNU-Linux system, because most Linux distributions include it. In this sense, installing, configuring, and maintaining SVN is a very simple task.
Most MS Windows users access the SVN server by the TortoiseSVN[18] client, because it provides the most usual interface for common users. Linux users usually use SVN utilities from the command-line, or eSvn[19]--a GUI frontend--with KDiff3[20] for showing complex differences.

56.5. Hosting LaTeX files in *Subversion*



agrarpol.bib 808 Jul 25 17:24

svn://svnAgrarKiel.dyndns.org/tex

Figure 200 Figure 1: Common `texmf` tree shown in *eSvns* ***Repository Browser***

On our *Subversion* server, we have one repository for a common `texmf` tree. Its structure complies with the **TeX Directory Structure** guidelines (TDS, `http://www.tug.org/tds/tds.html`, see figure 1). This repository provides LaTeX classes, LaTeX styles, and BibTeX styles that are not available in the LaTeX distributions of the users, e.g. because they were bought or developed for the internal use at our department. All users have a working copy of this repository and have configured LaTeX to use this as their personal `texmf` tree. For instance, teTeX (`http://www.tug.org/tetex/`) users can edit their TeX configuration file (e.g. `/etc/texmf/web2c/texmf.cnf`) and set the variable `TEXMFHOME` to the path of the working copy of the common `texmf` tree (e.g. by `TEXMFHOME = $HOME/texmf`); MiKTeX (`http://www.miktex.org/`) users can add the path of the working copy of the common `texmf` tree in the 'Roots' tab of the MiKTeX Options.
If a new class or style file has been added (but not if these files have been modified), the users have to update their 'file name data base' (FNDB) before they can use these classes

18 `http://tortoisesvn.tigris.org/`
19 `http://zoneit.free.fr/esvn/`
20 `http://kdiff3.sourceforge.net/`

and styles. For instance, teTeX users have to execute `texhash` ; MiKTeX users have to click on the button 'Refresh FNDB' in the 'General' tab of the MiKTeX Options.
Furthermore, the repository contains manuals explaining the specific LaTeX software solution at our department (e.g. this document).
The *Subversion* server hosts a separate repository for each project of our department. Although branching, merging, and tagging is less important for writing text documents than for writing source code for software, our repository layouts follow the recommendations of the 'Subversion book' (`http://svnbook.red-bean.com)`. In this sense, each repository has the three directories `/trunk` , `/branches` , and `/tags` .
The most important directory is `/trunk` . If a single text document belongs to the project, all files and subdirectories of this text document are in `/trunk` . If the project yields two or more different text documents, `/trunk` contains a subdirectory for each text document. A slightly different version (a **branch**) of a text document (e.g. for presentation at a conference) can be prepared either in an additional subdirectory of `/trunk` or in a new subdirectory of `/branches` . When a text document is submitted to a journal or a conference, we create a **`tag`** in the directory `/tags` so that it is easy to identify the submitted version of the document at a later date. This feature has been proven very useful. When creating branches and tags, it is important always to use the *Subversion* client (and not the tools of the local file system) for these actions, because this saves disk space on the server and it preserves information about the same history of these documents.
Often the question arises, which files should be put under version control. Generally, all files that are directly modified by the user and that are necessary for compiling the document should be included in the version control system. Typically, these are the LaTeX source code (`*.tex`) files (the main document and possibly some subdocuments) and all pictures that are inserted in the document (`*.eps` , `*.jpg` , `*.png` , and `*.pdf` files). All LaTeX classes (`*.cls`), LaTeX styles (`*.sty`), BibTeX data bases (`*.bib`), and BibTeX styles (`*.bst`) generally should be hosted in the repository of the common `texmf` tree, but they could be included in the respective repository, if some (external) co-authors do not have access to the common `texmf` tree. On the other hand, all files that are automatically created or modified during the compilation process (e.g. `*.aut` , `*.aux` , `*.bbl` , `*.bix` , `*.blg` , `*.dvi` , `*.glo` , `*.gls` , `*.idx` , `*.ilg` , `*.ind` , `*.ist` , `*.lof` , `*.log` , `*.lot` , `*.nav` , `*.nlo` , `*.out` , `*.pdf` , `*.ps` , `*.snm` , and `*.toc` files) or by the (LaTeX or BibTeX) editor (e.g. `*.bak` , `*.bib~` , `*.kilepr` , `*.prj` , `*.sav` , `*.tcp` , `*.tmp` , `*.tps` , and `*.tex~` files) generally should be **not** under version control, because these files are not necessary for compilation and generally do not include additional information. Furthermore, these files are regularly modified so that conflicts are very likely.

56.6. *Subversion* really makes the **diff** erence

A great feature of a version control system is that all authors can easily trace the workflow of a project by viewing the differences between arbitrary versions of the files. Authors are primarily interested in 'effective' modifications of the source code that change the compiled document, but not in 'ineffective' modifications that have no impact on the compiled document (e.g. the position of line breaks). Software tools for comparing text documents ('diff tools') generally cannot differentiate between 'effective' and 'ineffective' modifications; they highlight both types of modifications. This considerably increases the effort to find and review the 'effective' modifications. Therefore, 'ineffective' modifications should be avoided.

In this sense, it is very important not to change the positions of line breaks without cause. Hence, automatic line wrapping of the users' LaTeX editors should be turned off and line breaks should be added manually. Otherwise, if a single word in the beginning of a paragraph is added or removed, all line breaks of this paragraph might change so that most diff tools indicate the entire paragraph as modified, because they compare the files line by line. The diff tools *wdiff* (`http://www.gnu.org/software/wdiff/`) and *dwdiff* (`http://os.ghalkes.nl/dwdiff.html`) are not affected by the positions of line breaks, because they compare documents word by word. However, their output is less clear so that modifications are more difficult to track. Moreover, these tools cannot be used directly with the *Subversion* command-line switch `--diff-cmd` , but a small wrapper script has to be used (`http://textsnippets.com/posts/show/1033`).

A reasonable convention is to add a line break after each sentence and start each new sentence in a new line. Note that this has an advantage also beyond version control: if you want to find a sentence in your LaTeX code that you have seen in a compiled (DVI, PS, or PDF) file or on a printout, you can easily identify the first few words of this sentence and screen for these words on the left border of your editor window.

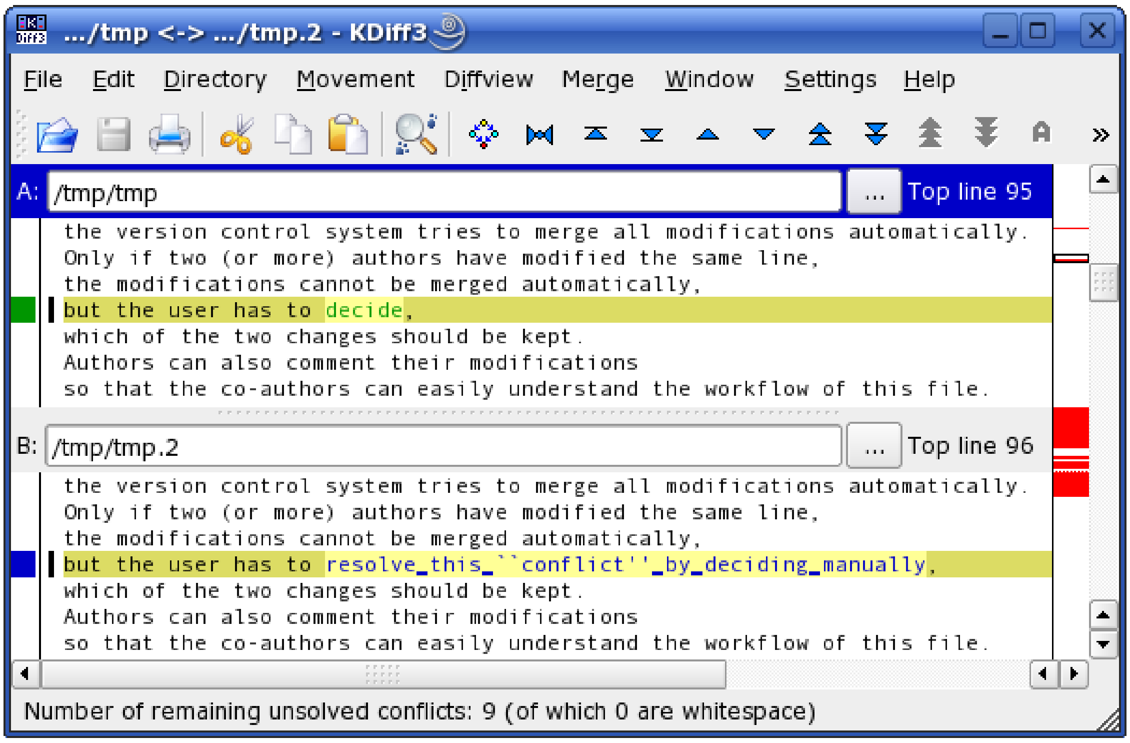
Furthermore, we split long sentences into several lines so that each line has at most 80 characters, because it is rather inconvenient to search for (small) differences in long lines. (Note: For instance, the LaTeX editor *Kile* (`http://kile.sourceforge.net/`) can assist the user in this task when it is configured to add a vertical line that marks the 80th column.) We find it very useful to introduce the additional line breaks at logical breaks of the sentence, e.g. before a relative clause or a new part of the sentence starts.

An example LaTeX code that is formatted according to these guidelines is the source code of the article *Tools for Collaborative Writing of Scientific LaTeX Documents* by Arne Henningsen[21] that is published (including the source code) in *The PracTeX Journal* 2007, Number 3 (`http://www.tug.org/pracjourn/2007-3/henningsen/`).

If the authors work on different operating systems, their LaTeX editors will probably save the files with different newline (end-of-line) characters (`http://en.wikipedia.org/wiki/Newline`). To avoid this type of 'ineffective' modifications, all users can agree on a specific newline character and configure their editor to use this newline character. Another alternative is to add the subversion property 'svn:eol-style' and set it to 'native'. In this case, *Subversion* automatically converts all newline characters of this file to the native newline character of the author's operating system (`http://svnbook.red-bean.com/en/1.4/svn.advanced.props.file-portability.html#svn.advanced.props.special.eol-style`).

There is also another important reason for reducing the number of 'ineffective' modifications: if several authors work on the same file, the probability that the same line is modified by two or more authors at the same time increases with the number of modified lines. Hence, 'ineffective' modifications unnecessarily increase the risk of conflicts (see section Interchanging Documents[22]).

21 `http://en.wikibooks.org/wiki/User%3AArnehe`
22 Chapter 56.3 on page 616


.../tmp <-> .../tmp.2 - KDiff3
File Edit Directory Movement Diffview Merge Window Settings Help
A: /tmp/tmp
Top line 95
B: /tmp/tmp.2
Top line 96

```
the version control system tries to merge all modifications automatically.
Only if two (or more) authors have modified the same line,
the modifications cannot be merged automatically,
but the user has to decide,
which of the two changes should be kept.
Authors can also comment their modifications
so that the co-authors can easily understand the workflow of this file.
```

```
the version control system tries to merge all modifications automatically.
Only if two (or more) authors have modified the same line,
the modifications cannot be merged automatically,
but the user has to resolve_this_``conflict''_by_deciding_manually,
which of the two changes should be kept.
Authors can also comment their modifications
so that the co-authors can easily understand the workflow of this file.
```

Number of remaining unsolved conflicts: 9 (of which 0 are whitespace)

Figure 201 Figure 2: Reviewing modifications in *KDiff3*

Furthermore, version control systems allow a very effective quality assurance measure: all authors should critically review their own modifications before they commit them to the repository (see figure 2). The differences between the user's working copy and the repository can be easily inspected with a single *Subversion* command or with one or two clicks in a graphical *Subversion* client. Furthermore, authors should verify that their code can be compiled flawlessly before they commit their modifications to the repository. Otherwise, the co-authors have to pay for these mistakes when they want to compile the document. However, this directive is not only reasonable for version control systems but also for all other ways to interchange documents among authors.
Subversion has a feature called 'Keyword Substitution' that includes dynamic version information about a file (e.g. the revision number or the last author) into the contents of the file itself (see e.g. `http://svnbook.red-bean.com`, chapter 3). Sometimes, it is useful to include these information not only as a comment in the LaTeX source code, but also in the (compiled) DVI, PS, or PDF document. This can be achieved with the LaTeX packages *svn* (`http://www.ctan.org/tex-archive/macros/latex/contrib/svn/`), *svninfo* (`http://www.ctan.org/tex-archive/macros/latex/contrib/svninfo/`), or (preferably) *svn-multi* (`http://www.ctan.org/tex-archive/macros/latex/contrib/svn-multi/`).
The most important directives for collaborative writing of LaTeX documents with version control systems are summarised in the following box.

Directives for using LaTeX with version control systems

1. Avoid 'ineffective' modifications.
2. Do not change line breaks without good reason.
3. Turn off automatic line wrapping of your LaTeX editor.
4. Start each new sentence in a new line.

5. Split long sentences into several lines so that each line has at most 80 characters.
6. Put only those files under version control that are directly modified by the user.
7. Verify that your code can be compiled flawlessly before committing your modifications to the repository.
8. Use *Subversion*s ***diff* feature to critically review your modifications before committing them to the repository.**
9. Add a meaningful and descriptive comment when committing your modifications to the repository.
10. Use the *Subversion* client for copying, moving, or renaming files and folders that are under revision control.

If the users are willing to let go of the built-in *diff* utility of SVN and use *diff* tools that are local on their workstations, they can put to use such tools that are more tailored to text documents. The *diff* tool that comes with SVN was designed with source code in mind. As such, it is built to be more useful for files of short lines. Other tools, such as **Compare It!** allows to conveniently compare text files where each line can span hundreds of characters (such as when each line represents a paragraph). When using a *diff* tool that allows convenient views of files with long lines, the users can author the TeX files without a strict line-breaking policy.

56.6.1. Visualizing *diffs* in LaTeX: latexdiff and changebar

The tools latexdiff[23] and changebar[24] can visualize differences of two LaTeX files inside a generated document. This makes it easier to see impact of certain changes or discuss changes with people not custom to LaTeX. Changebar comes with a script `chbar.sh` which inserts a bar in the margin indicating parts that have changed. Latexdiff allows different styles of visualization. The default is that discarded text is marked as red and added text is marked as blue. It also supports a mode similar to Changebar which adds a bar in the margin. Latexdiff comes with a script `latexrevise` which can be used to accept or decline changes. It also has a wrapper script to support version control systems such as the discussed Subversion.
An example on how to use Latexdiff in the Terminal.

```
   latexdiff old.tex new.tex > diff.tex              # Files old.tex and
new.tex are compared and the file visualizing the changes is written to diff.tex
   pdflatex diff.tex                                 # Create a PDF showing
the changes
```

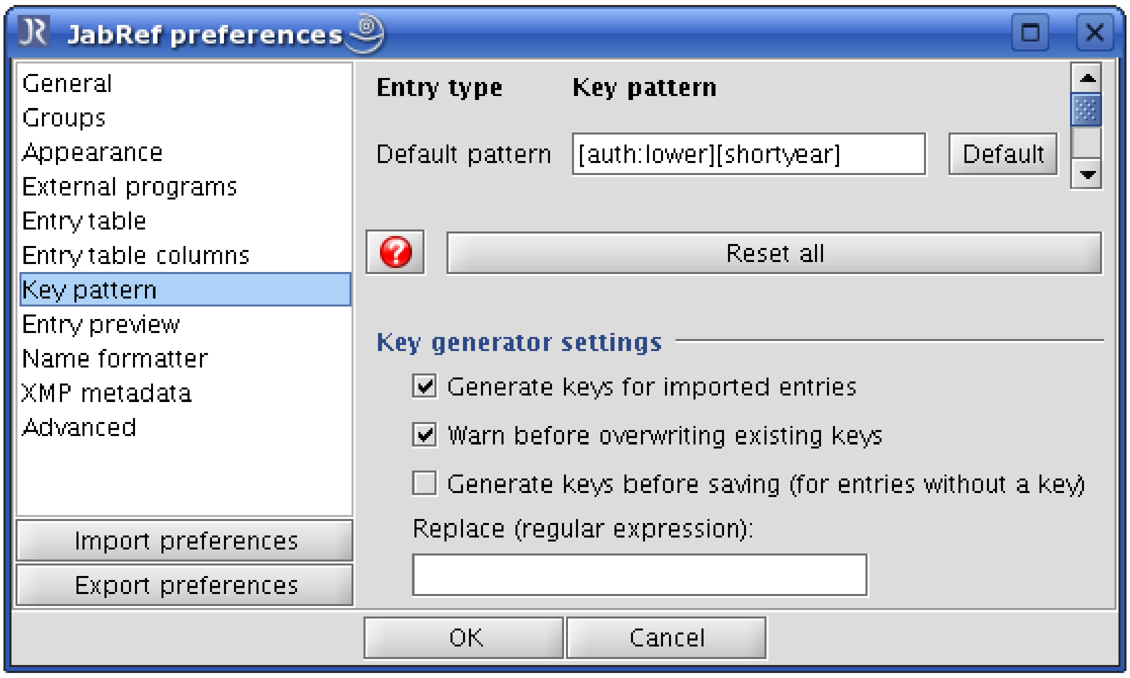
The program DiffPDF[25] can be used to compare two existing PDFs visually. There is also a command line tool comparepdf[26] based on DiffPDF.

56.7. Managing collaborative bibliographies

Writing of scientific articles, reports, and books requires the citation of all relevant sources. BibTeX is an excellent tool for citing references and creating bibliographies

23 `http://www.ctan.org/tex-archive/support/latexdiff/`
24 `http://www.ctan.org/tex-archive/macros/latex/contrib/changebar/`
25 `http://www.qtrac.eu/diffpdf.html`
26 `http://www.qtrac.eu/comparepdf.html`

(Markey 2005, Fenn 2006). Many different BibTeX styles can be found on CTAN (`http://www.ctan.org`) and on the LaTeX Bibliography Styles Database (`http://jo.irisson.free.fr/bstdatabase/`). If no suitable BibTeX style can be found, most desired styles can be conveniently assembled with *custombib* /*makebst* (`http://www.ctan.org/tex-archive/macros/latex/contrib/custom-bib/`). Furthermore, BibTeX style files can be created or modified manually; however this action requires knowledge of the (unnamed) postfix stack language that is used in BibTeX style files (Patashnik 1988).
At our department, we have a common bibliographic data base in the BibTeX format (.bib file). It resides in our common `texmf` tree (see section 'Hosting LaTeX files in *Subversion)* in the subdirectory `/bibtex/bib/` (see figure 1). Hence, all users can specify this bibliography by only using the file name (without the full path) --- no matter where the user's working copy of the common `texmf` tree is located.
All users edit our bibliographic data base with the graphical BibTeX editor *JabRef* (`http://jabref.sourceforge.net/`). As *JabRef* is written in *Java* , it runs on all major operating systems. As different versions of *JabRef* generally save files in a slightly different way (e.g. by introducing line breaks at different positions), all users should use the same (e.g. last stable) version of *JabRef* . Otherwise, there would be many differences between different versions of `.bib` files that solely originate from using different version of *JabRef* . Hence, it would be hard to find the real differences between the compared documents. Furthermore, the probability of conflicts would be much higher (see section 'Subversion really makes the difference'). As *JabRef* saves the BibTeX data base with the native newline character of the author's operating system, it is recommended to add the *Subversion* property 'svn:eol-style' and set it to 'native' (see section 'Subversion really makes the difference').



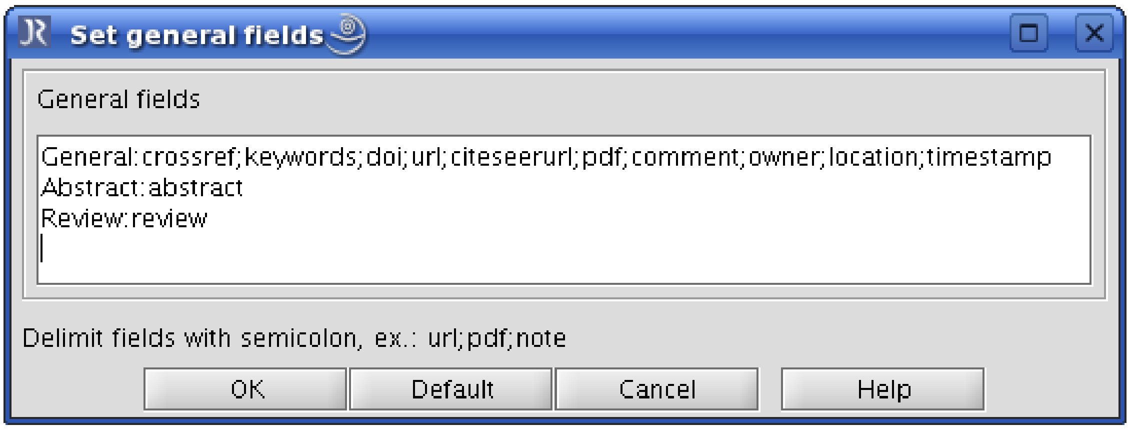
Default pattern: [auth:lower][shortyear]

Key generator settings
- [✓] Generate keys for imported entries
- [✓] Warn before overwriting existing keys
- [ ] Generate keys before saving (for entries without a key)

Replace (regular expression):

Figure 202 Figure 3: Specify default key pattern in *JabRef*

JabRef is highly flexible and can be configured in many details. We make the following changes to the default configuration of *JabRef* to simplify our work. First, we specify the default pattern for BibTeX keys so that *JabRef* can automatically generate keys in our desired format. This can be done by selecting `Options` → `Preferences` → `Key pattern` and modifying the desired pattern in the field `Default pattern` . For instance, we use `[auth:lower][shortyear]` to get the last name of the first author in lower case and the last two digits of the year of the publication (see figure 3).

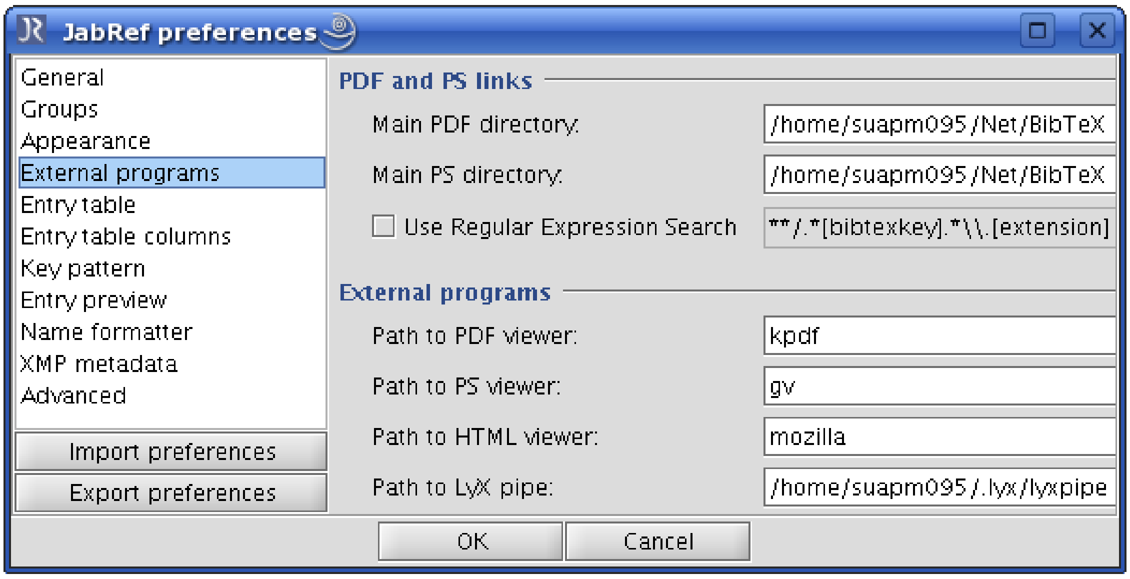


General:crossref;keywords;doi;url;citeseerurl;pdf;comment;owner;location;timestamp
Abstract:abstract
Review:review

Delimit fields with semicolon, ex.: url;pdf;note

Figure 203 Figure 4: Set up general fields in *JabRef*

Second, we add the BibTeX field `location` for information about the location, where the publication is available as hard copy (e.g. a book or a copy of an article). This field can contain the name of the user who has the hard copy and where he has it or the name of a library and the shelf-mark. This field can be added in *JabRef* by selecting `Options` → `Set up general fields` and adding the word `location` (using the semicolon (`;`) as delimiter) somewhere in the line that starts with `General:` (see figure 4).



Main PDF directory: /home/suapm095/Net/BibTeX
Main PS directory: /home/suapm095/Net/BibTeX
☐ Use Regular Expression Search **/.*[bibtexkey].*\\.[extension]

Path to PDF viewer: kpdf
Path to PS viewer: gv
Path to HTML viewer: mozilla
Path to LyX pipe: /home/suapm095/.lyx/lyxpipe

Figure 204 Figure 5: Specify 'Main PDF directory' in *JabRef*

Third, we put all PDF files of publications in a specific subdirectory in our file server, where we use the BibTeX key as file name. We inform *JabRef* about this subdirectory by selecting `Options` → `Preferences` → `External programs` and adding the path of the this subdirectory in the field `Main PDF directory` (see figure 5). If a PDF file of a

publication is available, the user can push the `Auto` button left of *JabRefs* ***`Pdf` field*** to automatically add the file name of the PDF file. Now, all users who have access to the file server can open the PDF file of a publication by simply clicking on *JabRefs* ***PDF icon.***
If we send the LaTeX source code of a project to a journal, publisher, or somebody else who has no access to our common `texmf` tree, we do not include our entire bibliographic data base, but extract the relevant entries with the Perl script *aux2bib* (`http://www.ctan.org/tex-archive/biblio/bibtex/utils/bibtools/aux2bib`).

56.8. Conclusion

This wikibook describes a possible way to efficiently organise the collaborative preparation of LaTeX documents. The presented solution is based on the *Subversion* version control system and several other software tools and LaTeX packages. However, there are still a few issues that can be improved.
First, we plan that all users install the same LaTeX distribution. As the *TeX Live* distribution (`http://www.tug.org/texlive/`) is available both for Unix and MS Windows operating systems, we might recommend our users to switch to this LaTeX distribution in the future. (Currently, our users have different LaTeX distributions that provide a different selection of LaTeX packages and different versions of some packages. We solve this problem by providing some packages on our common `texmf` tree.)
Second, we consider to simplify the solution for a common bibliographic data base. Currently it is based on the version control system *Subversion* , the graphical BibTeX editor *JabRef* , and a file server for the PDF files of publications in the data base. The usage of three different tools for one task is rather challenging for infrequent users and users that are not familiar with these tools. Furthermore, the file server can be only accessed by local users. Therefore, we consider to implement an integrated server solution like *WIKINDX* (`http://wikindx.sourceforge.net/`), *Aigaion* (`http://www.aigaion.nl/`), or *refBASE* (`http://refbase.sourceforge.net/`). Using this solution only requires a computer with internet access and a web browser, which makes the usage of our data base considerably easier for infrequent users. Moreover, the stored PDF files are available not only from within the department, but throughout the world. (Depending on the copy rights of the stored PDF files, the access to the server --- or least the access to the PDF files --- has to be restricted to members of the department.) Even Non-LaTeX users of our department might benefit from a server-based solution, because it should be easier to use this bibliographic data base in (other) word processing software packages, because these servers provide the data not only in BibTeX format, but also in other formats.
All readers are encouraged to contribute to this wikibook by adding further hints or ideas or by providing further solutions to the problem of collaborative writing of LaTeX documents.

56.9. Acknowledgements

Arne Henningsen thanks Francisco Reinaldo and Géraldine Henningsen for comments and suggestions that helped him to improve and clarify this paper, Karsten Heymann for many hints and advices regarding LaTeX, BibTeX, and *Subversion* , and Christian

Henning as well as his colleagues for supporting his intention to establish LaTeX and *Subversion* at their department.

56.10. References

- Fenn, Jürgen (2006): Managing citations and your bibliography with BibTeX. The PracTEX Journal, 4. `http://www.tug.org/pracjourn/2006-4/fenn/`.
- Markey, Nicolas (2005): Tame the BeaST. The B to X of BibTeX. `http://www.ctan.org/tex-archive/info/bibtex/tamethebeast/ttb_en.pdf`. Version 1.3.
- Oren Patashnik. Designing BibTeX styles. `http://www.ctan.org/tex-archive/info/biblio/bibtex/contrib/doc/btxhak.pdf`.
- Tools for collaborative paper-writing[27]

27 `http://mathoverflow.net/questions/3044/tools-for-collaborative-paper-writing`

57. Export To Other Formats

Strictly speaking, LaTeX source can be used to directly generate two formats:
• DVI using `latex` , the first one to be supported;
• PDF using `pdflatex`, more recent.
Using other software freely available on Internet, you can easily convert DVI and PDF to other document formats. In particular, you can obtain the PostScript version using software which is included in your LaTeX distribution. Some LaTeX IDE will give you the possibility to generate the PostScript version directly (even if it uses internally a DVI mid-step, e.g. LaTeX $\rightarrow$ DVI $\rightarrow$ PS). It is also possible to create PDF from DVI and vice versa. It doesn't seem logical to create a file with two steps when you can create it straight away, but some users might need it because, as you remember from the first chapters, the format you can generate depends upon the formats of the images you want to include (EPS for DVI, PNG and JPG for PDF). Here you will find sections about different formats with description about how to get it.
Other formats can be produced, such as RTF (which can be used in Microsoft Word) and HTML. However, these documents are produced from software that parses and interprets the LaTeX files, and do not implement all the features available for the primary DVI and PDF outputs. Nonetheless, they do work, and can be crucial tools for collaboration with colleagues who do not edit documents with LaTeX.

57.1. Tools installation

This chapter features a lot of third-party tools; most of them are installed independently of your TeX distribution.
Some tools are Unix-specific (*BSD, GNU/Linux and Mac OS X), but it may be possible to make them work on Windows. If you have the choice, it is often easier with Unix systems for command line tools.
Some tools may already be installed. For instance, you can check if dvipng is installed and ready to use (Unix only):

```
which dvipng
```

You get a directory if it is OK. [[w Most of these tools are installable using your package manager or portage tree (Unix only).

57.2. Preview mode

This section describes how to generate a screenshot of a LaTeX page or of a specific part of the page using the LaTeX package `preview`. Screenshots are useful, for example, if you want to include a LaTeX generated formula on a presentation using you favorite slideware

like Powerpoint, Keynote or LibreOffice Impress. First, start by making sure you have `preview`. See Installing Extra Packages[1].
Say you want to take a screenshot of

$$\pi = \sqrt{12}\sum^\infty_{k=0} \frac{(-3)^{-k}}{2k+1}.$$

Write this formula in the `preview` environment:

```
\documentclass{article}
\usepackage[active]{preview}
\begin{document}
\begin{preview}
\[
\pi = \sqrt{12}\sum^\infty_{k=0} \frac{ (-3)^{-k} }{ 2k+1 }
\]
\end{preview}
\end{document}
```

Note the `active` option in the package declaration and the `preview` environment around the equation's code. Without any of these two, you won't get any output.
This package is also very useful to export specific parts to other format, or to produce graphics (*e.g.* using PGF/TikZ[2]) and then including them in other documents. You can also automate the previewing of specific environments:

```
\usepackage[active,tightpage]{preview}
\PreviewEnvironment{lstlisting}
\setlength{\PreviewBorder}{10pt}%

% ...

\begin{lstlisting}
int main()
{
        /* ... */
}
\end{lstlisting}
```

This will produce a PDF containing only the listing content, the *page* layout will depend on the shape of the source code.

57.3. Convert to PDF

57.3.1. Directly

```
pdflatex my_file
```

57.3.2. DVI to PDF

```
dvipdfm my_file.dvi
```

will create `my_file.pdf` . Another way is to pass through PS generation:

```
dvi2ps myfile.dvi
ps2pdf myfile.ps
```

1 Chapter 3 on page 29
2 Chapter 47 on page 535

you will get also a file called *my_file.ps* that you can delete.

57.3.3. Merging PDF

If you have created different PDF documents and you want to merge them into one single PDF file you can use the following command-line command. You need to have Ghostscript installed:

Using Windows

```
gswin32 -dNOPAUSE -sDEVICE=pdfwrite -sOUTPUTFILE=Merged.pdf -dBATCH 1.pdf 2.pdf
3.pdf
```

Using Linux

```
gs -dNOPAUSE -sDEVICE=pdfwrite -sOUTPUTFILE=Merged.pdf -dBATCH 1.pdf 2.pdf
3.pdf
```

Alternatively, PDF-Shuffler[3] is a small python-gtk application, which helps the user to merge or split pdf documents and rotate, crop and rearrange their pages using an interactive and intuitive graphical interface. This program may be avaliable in your Linux distribution's repository.
Another option to check out is pdftk[4] (or PDF toolkit), which is a command-line tool that can manipulate PDFs in many ways. To merge one or more files, use:

```
pdftk 1.pdf 2.pdf 3.pdf cat output 123.pdf
```

Using pdfLaTeX

Note: If you are merging external PDF documents into a LaTeX document which is compiled with `pdflatex` , a much simpler option is to use the `pdfpages` package, *e.g.* :

```
\usepackage{pdfpages}
...
\includepdf[pages=-]{Document1.pdf}
\includepdf[pages=-]{Document2.pdf}
...
```

Three simple shell[5] scripts using the `pdfpages` package are provided in the pdfjam bundle[6] by D. Firth. They include options to merge several pdf files (pdfjoin), put several pages in one physical sheet (pdfnup) and rotate pages (pdf90).
See also Modular Documents[7]

3 http://pdfshuffler.sourceforge.net/
4 http://www.accesspdf.com/
5 http://en.wikipedia.org/wiki/Shell%20%28computing%29
6 http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic/firth/software/pdfjam
7 Chapter 55.2.6 on page 612

57.3.4. XeTeX

You can also use XeTeX (or, more precisely, XeLaTeX), which works in the same way as `pdflatex` : it creates a PDF file directly from LaTeX source. One advantage of XeTeX over standard LaTeX is support for Unicode and modern typography. See its Wikipedia entry[8] for more details.
Customization of PDF output in XeTeX (setting document title, author, keywords etc.) is done using the configuration of hyperref[9] package.

57.4. Convert to PostScript

from PDF

```
pdf2ps my_file.pdf
```

from DVI

```
dvi2ps my_file.dvi
```

57.5. Convert to RTF

LaTeX can be converted into an RTF file, which in turn can be opened by a word processor such as LibreOffice Writer[10] or Microsoft Word[11]. This conversion is done through latex2rtf[12], which may run on any computer platform, however is only actively supported on Windows, Linux and BSD, with the last mac update being from 2001. The program operates by reading the LaTeX source, and mimicking the behaviour of the LaTeX program. `latex2rtf` supports most of the standard implementations of LaTeX, such as standard formatting, some math typesetting, inclusion of EPS, PNG or JPG graphics, and tables. As well, it has some limited support for packages, such as varioref, and natbib. However, many other packages are not supported.
`latex2rtf` is simple to use. The Windows version has a GUI (`l2rshell.exe`), which is straightforward to use. The command-line version is offered for all platforms, and can be used on an example `mypaper.tex` file:

```
latex mypaper
bibtex mypaper # if you use bibtex
latex2rtf mypaper
```

Both `latex` and (if needed) `bibtex` commands need to be run *before* `latex2rtf` , because the `.aux` and `.bbl` files are needed to produce the proper output. The result of this conversion will create `myfile.rtf` , which you may open in many word processors such as Microsoft Word or LibreOffice.

8 `http://en.wikipedia.org/wiki/XeTeX`
9 Chapter 20.3 on page 258
10 `http://en.wikipedia.org/wiki/LibreOffice%20Writer`
11 `http://en.wikipedia.org/wiki/Microsoft%20Word`
12 `http://latex2rtf.sourceforge.net/`

57.6. Convert to HTML

There are many converters to HTML.
HEVEA[13]

```
hevea mylatexfile
```

latex2html

```
latex2html -html_version 4.0,latin1,unicode -split 1 -nonavigation -noinfo
-title "MyDocument" MyDocument.tex
```

LaTeXML[14]

```
latexmlc paper.tex --destination=paper.html
```

pdf2htmlEX[15]

```
pdf2htmlEX [options] <input.pdf> [<output.html>]
```

pdf2htmlEX can convert PDF to HTML without losing text or format. It is designed as a general PDF to HTML converter, not only restricted to the PDF generated by LaTeX source. LaTeX users can compile the LaTeX source code to PDF, and then convert the PDF to HTML via pdf2htmlEX. Some introductions of pdf2htmlEX can be found on its own wiki page[16]. More technical details can be found on the paper published on TUGboat: *Online publishing via pdf2htmlEX* HTML[17] / PDF[18]. The Figure 3 of the paper gives different work-flows of publishing HTML online.
TeX4ht
TeX4ht[19] is a very powerful conversion program, but its configuration is not straightforward. Basically a configuration file has to be prepared, and then the program is called.
bibtex2html
For BibTeX.

```
bibtex2html mybibtexfile
```

57.7. Convert to image formats

It is sometimes useful to convert LaTeX output to image formats for use in systems that do not support DVI nor PDF files, such as Wikipedia.

13 http://hevea.inria.fr
14 http://en.wikipedia.org/wiki/LaTeXML
15 https://github.com/coolwanglu/pdf2htmlEX
16 https://github.com/coolwanglu/pdf2htmlEX/wiki
17 http://coolwanglu.github.io/pdf2htmlEX/doc/tb108wang.html
18 http://coolwanglu.github.io/pdf2htmlEX/doc/tb108wang.pdf
19 http://www.cse.ohio-state.edu/~gurari/TeX4ht/

There are two family of graphics:

- Vector graphics can be scaled to any size, thus do not suffer from quality loss. SVG[20] is a vector format.
- Raster graphics define every pixel explicitly. PNG[21] is a raster format.

So vector graphics are usually preferred. There is still some cases where raster graphics are used:

- The target system does not handle vector graphics, only raster graphics are supported.
- SVG can not embed fonts. So either the font will be rendered using a local .ttf or .otf font (which will mostly change the output), or all characters must be turned to vector graphics. This last method makes the SVG big and slow. If the input LaTeX file contains a lot of text which formatting must be preserved, SVG is not that great.

So SVG is great for drawings and a small amount of text. JPG is a well known raster formats, however it is usually not as good as PNG for text.
In some cases it may be sufficient to simply copy a region of a PDF (or PS) file using the tools available in a PDF viewer (for example using LaTeX to typeset a formula for pasting into a presentation). This however will not generally have sufficient resolution for whole pages or large areas.

57.7.1. Multiple formats

pdftocairo
There is `pdftocairo` featured in the poppler toolset.

```
pdftocairo -svg latexdoc.pdf output.svg
```

`pdftocairo` also supports various raster graphic formats.

57.7.2. Vector graphics

pdf2svg
Direct conversion from PDF to SVG can be done using the command line tool pdf2svg[22].

```
pdf2svg file.pdf file.svg
```

ps2svg
Alternatively DVI or PDF can be converted to PS as described before, then the bash script ps2svg.sh[23] can be used (as all the software used by this script is multiplatform, this is also possible in Windows, a step-by-step guide could be written).
dvisvgm
One can also use dvisvgm[24], an open source utility that converts from DVI to SVG.

20 `http://en.wikipedia.org/wiki/Scalable%20Vector%20Graphics`
21 `http://en.wikipedia.org/wiki/PNG`
22 `http://www.cityinthesky.co.uk/opensource/pdf2svg/`
23 `http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Electronics/Ps2svg.sh`
24 `http://dvisvgm.sourceforge.net/`

```
dvisvgm -n file.dvi
```

Inkscape
Inkscape is able to convert to SVG, PDF, EPS, and other vector graphic formats.

```
inkscape --export-area-drawing --export-ps=OUTPUT INPUT
inkscape --export-area-page --export-plain-svg=OUTPUT INPUT
```

57.7.3. Raster graphics

GIMP
Open your file with GIMP[25]. It will ask you which page you want to convert, whether you want to use anti-aliasing (choose *strong* if you want to get something similar to what you see on the screen). Try different resolutions to fit your needs, but 100 dpi should be enough. Once you have the image within GIMP, you can post-process it as you like and save it to any format supported by GIMP, as PNG for example.
dvipng
A method for DVI files is dvipng[26]. Usage is the same as `dvipdfm` .
Run `latex` as usual to generate the dvi file. Now, we want an X font size formula, where X is measure in pixels. You need to convert this, to dots per inch (dpi). The formula is: `<dpi> = <font_px>*72.27/10` . If you want, for instance, X = 32, then the size in dpi corresponds to 231.26. This value will be passed to `dvipng` using the flag `-D` . To generate the desired png file run the command as follows:

```
dvipng -T tight -D 231.26 -o foo.png foo.dvi
```

The flag `-T` sets the size of the image. The option `tight` will only include all ink put on the page. The option `-o` sends the output to the file name `foo.png` .
ImageMagick
The `convert` command from the ImageMagick[27] suite can convert both DVI and PDF files to PNG.

```
convert input.pdf output.png
```

optipng
You can optimize the resulting image using optipng[28] so that it will take up less space.

57.8. Convert to plain text

If you are thinking of converting to plain text for spell-checking or to count words, there may be an easier way -- read Tips and Tricks[29] first.

25 `http://en.wikibooks.org/wiki/GIMP`
26 `http://savannah.nongnu.org/projects/dvipng/`
27 `http://www.imagemagick.org/`
28 `http://optipng.sourceforge.net/`
29 Chapter 59.6 on page 648

Most LaTeX distributions come with `detex` program, which strips LaTeX commands. It can handle multi-file projects, so all you need is to give one command:

```
detex yourfile
```

(note the omission of .tex extension). This will output result to standard output. If you want the plain text go to a file, use

```
detex yourfile > yourfile.txt
```

If the output from `detex` does not satisfy you, you can try a newer version available on Google Code[30], or use HTML conversion first and then copy text from your browser.
If you want to keep the formating, you can use a *DVI-to-plain text* converter, like `catdvi` . Example:

```
catdvi yourfile.dvi | fmt -u
```

The use of *fmt -u* (available on most Unices) will remove the justification.

30 `http://code.google.com/p/opendetex/`

Part X.

Help and Recommendations

58. FAQ

58.1. Margins are too wide

LaTeX's default margins may seem too large. In most cases, this is a preferred default and improves readability.
If you still disagree, you can easily change them with

```
\usepackage{geometry}
% or
\usepackage[margin=1.5in]{geometry}
```

See Page Layout[1].

58.2. Avoid excessive double line breaks in source code

Too many paragraphs of one line or two do not look very good.
Remember the TeX rule:

- If two or more consecutive line breaks are found, TeX starts a new paragraph.
- If only one linebreak is found, TeX inserts a space if there is no space directly before or after it.

You might be tempted to put blank lines all the time to improve the readability of your source code, but this will have an impact on formatting. The solution is simple: put a comment at the very beginning of the blank lines. This will prevent TeX from seeing another line break—all characters up to and including the next line break after a comment are ignored.
Example:

```
We are in the first paragraph here.
%
We are still in the first paragraph.

This time, this is another paragraph.
```

58.3. Simplified special character input

So long as your computing environment supports UTF-8, you can enter special characters directly rather than entering the TeX commands for diacritics and other extended characters. E.g.,

```
R\'esum\'e can also be written résumé.
```

This requires that:

- your text editor supports and is set to save your file in UTF-8;
- you add the `\usepackage[utf8]{inputenc}` line in the preamble.

Avoid using `latin1`. See Special Characters[2].

1 Chapter 16 on page 193
2 Chapter 11 on page 123

58.4. Writing the euro symbol directly

Add the following lines in your preamble:

```
\usepackage[utf8]{inputenc}
\usepackage{marvosym}
\DeclareUnicodeCharacter{20AC}{\EUR{}}
```

58.5. LaTeX paragraph headings have title and content on the same line

Some people do not like the way `\paragraph{...}` writes the title on the same line as the content. This is actually fairly common in a lot of documents and not as weird as it may seem at first.
There are ways to get around the default behavior, however; see \paragraph line break[3] for more information.

58.6. *Fonts are ugly/jagged/bitmaps* or *PDF search fails* or *Copy/paste from PDF is messy*

You must be using diacritics (*e.g.* accents) with OT1 encoding (the default). Switch to T1 encoding:

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

If you have ugly jagged fonts after the font encoding change, then you have no Type1 compatible fonts available. Install `Computer Modern Super` or `Latin Modern` (package name may be `lm`). To use `Latin Modern` you need to include the package:

```
\usepackage{lmodern}
```

See Fonts[4] for an explanation.

58.7. Manual formatting: use of line breaks and page breaks

You should *really* avoid breaking lines and pages manually. The TeX engine is in charge of that. The big problem with manual formatting is that it is not dynamic. Even if it looks right the first time, the content is likely to render really badly if you change anything before the point you manually formatted.
The only place where page breaks are recommended is at the upper level of sectioning in your documents, *e.g.* parts or chapters (although when you start a new part or chapter, LaTeX will ordinarily do this for you). When you do manually insert a page break, you should use `\clearpage` or `\cleardoublepage` which print currently floating figures before starting a new page.
If you absolutely have to insert line or page breaks manually, you should do it after you are sure you have completed your document otherwise, so that you don't later have to come back and update it.

3 Chapter 7.6.1 on page 85
4 Chapter 9.5 on page 99

58.8. Always finish commands with *{}*

TeX has an unintuitive rule that if a control sequence (a command) is not followed by a pair of braces (with a parameter in between or not), then the following space character(s) are ignored. LaTeX will not print *any* space, and the command (say, the TeX or LaTeX logos) are run together with the following word.
To fix this, use a pair of braces after the command, even if there are no parameters. Example:

```
\LaTeX is great. % BAD !
\LaTeX{} is great. % GOOD !
```

(Technical explanation: a control sequence name can only be composed of characters with catcode 11, that is A-Z and a-z by default. TeX knows where the control sequence name start thanks to the backslash, and it knows where it ends when it encounters the first token which is not of catcode 11. This character is then skipped. Since consecutive spaces have been concatenated into one single space, no space is taken into account.)
It is possible to define macros that will insert a space dynamically by using the `xspace` package.

- If there is no brace and a space following the command, an extra space will be appended.
- If there are braces, no extra space will be printed.

Example:

```
\usepackage{xspace}
\let\latexold\LaTeX
\renewcommand{\LaTeX}{\textrm{\latexold}\xspace}
...
\LaTeX is followed by a space.
\LaTeX{} is followed by a space.
\LaTeX{}is not followed by a space.
```

58.9. Avoid bold and underline

Typographically speaking, it is usually poor practice to use bold or underline formats in the middle of a paragraph. This has become a common habit for users of traditional word processors because these two functions are very easily accessible (along with italics).
However, bold and underline tend to overweight the text and distract the reader. When you start reading a paragraph with a bold word in the middle, you often read the emphasized part first, thus spoiling the content and breaking the order of the ideas. Italics are less obvious and do not have more weight than normal characters, so they are usually a better choice for emphasizing small amounts of text.
The original and more appropriate use of bold and underline is for special parts, such as headers, the index, glossaries, and so on. (Actually, underlining is rarely used in professional environments.)
LaTeX has a macro `\emph{...}` for emphasizing text using italics. It should be preferred to `\textit{...}` because `\emph{...}` will correctly print emphasized text inside other italic text in the regular font.

58.10. The proper way to use figures

Users used to WYSIWYG document processors like Microsoft Word or LibreOffice often get frustrated with figures. The answer is simple: a figure is *not* a picture!

If you use `\includegraphics` without enclosing it in a `figure` environment, it will behave just as in a word processor, placing the picture right at the spot where it was placed in the source.
Figures are a type of float, which is a virtual object that LaTeX can put in places other than the exact location it was created, which helps to prevent cluttering your text with pictures and tables.
See Importing Graphics[5] and Floats, Figures and Captions[6] for more details.

58.11. Text stops justifying

Most likely you have used `\raggedleft`, `\raggedright` or `\centering` at some point and forgotten to switch it off. These commands are switches—they remain active until the end of the scope, or until the end of the document if there is no scope. See Paragraph Alignment[7] for more information.

58.12. Rules of punctuation and spacing

LaTeX does some work for you, but not everything. Especially regarding punctuation, you are pretty free to do what you want. Punctuation rules are different for each language. In English there is *no* space before a punctuation mark and one space after it.
There are a lot of rules, but you can have a quick look at Wikipedia[8].

58.13. Compilation fails after a Babel language change

This is a limitation of Babel. Delete the `.aux` file (or clean the project), then try compiling again.

58.14. Learning LaTeX quickly or correctly

Nowadays it is very common to "learn" on the web by using a search engine and copying and pasting things here and there. As with every programming language, this is generally a poor method which will lead to lack of control, unexpected results, and a lot of frustration. Really learning LaTeX is not that difficult and does not take that long. Most chapters in this book are dedicated to a specific usage, so the basics are actually covered very quickly.
If you are getting frustrated with a specific package, make sure you read its official documentation, which is usually the best source of information. Content found on the web, even in this book, is rarely as accurate as the official documentation. Inaccurate information might result in causing mistakes without you understanding why.
The time you spend learning is worth it, and it quickly makes up for the time you would lose if you don't learn things properly and end up stuck all the time.

5 Chapter 17 on page 211
6 Chapter 18 on page 231
7 Chapter 7.1 on page 79
8 `http://en.wikipedia.org/wiki/Punctuation%23Conventional_styles_of_English_punctuation`

58.15. Non-breaking spaces

This useful feature is unknown to most newcomers, although it is available on most WYSIWYG document processors. A non-breaking space between two tokens (e.g. words, punctuation marks) prevents processors from inserting a line break between them. It is very important for consistent reading. LaTeX uses the '~' symbol as a non-breaking space.
You usually use non-breaking spaces for punctuation marks in some languages, for units and currencies, for initials, etc.
For example, in French typography, you put a non-breaking space before all two-parts punctuation marks. Example:

```
Il répondit~: «~Ce pain coûte-t-il 2~€~?~»
```

Note that writing French like this might get really painful. Thankfully, Babel with the `frenchb` option will take care of the non-breaking spaces for all punctuation marks. In the above example, only the non-breaking space for the euro symbol must remain.

58.16. Smart mathematics

All virtual objects designated by letters, variables or others should use a dedicated formatting. For math and a lot of other fields, the LaTeX math formatting is perfect. For instance, if you want to refer to an object A , write

```
Speaking of $A$, let's say...
```

If you want to refer to several objects in a sentence, it is the same.

```
Speaking of $A$, $B$ and $C$...
```

If you refer to a set of objects, you can still use the math notation.

```
The family $(A, B, C)$ is...
```

Note that this is different from usual text parentheses.

```
A sentence. ($A$, $B$, and $C$ are not concerned, but we do not mean the $(A, B,
 C)$ family.)
```

58.17. Use vector graphics rather than raster images

Raster (bitmap) graphics scale poorly and often create jagged or low-quality results which clash with the document quality, particularly when printed.
Using vector (line-oriented) graphics instead, either through LaTeX's native diagramming tools or by exporting vector formats from your drawing or diagramming tools, will produce much higher quality results. When possible, you should prefer PDF, EPS, or SVG graphics over PNG or JPG.

58.18. Stretching tables

Trying to stretch tables with the default `tabular` environment will often lead to unexpected results. The nice `tabu` package will do what you want and even much more.

Alternatively if you cannot use the `tabu` package you may try `tabularx` or `tabulary` packages See Tables[9].

58.19. Tables are easier than you think

Even though the Tables[10] chapter is quite long, it is worth reading. In the end, you only need to know a few things about the environment of your choice.
Some LaTeX editors feature table assistants. Also, many spreadsheet applications have a LaTeX export feature (or plugin). Again, see Tables[11] for more details.

58.20. Relieving cumbersome code (lists and long command names)

LaTeX is sometimes cumbersome to write, especially if you are not using an adequate editor. See Editors[12] for some interesting choices.
You can define aliases to shorten some commands:

```
\usepackage{xspace}
\newcommand\tss[1]{\textsuperscript{#1}}
\newcommand\tbs[1]{\textbackslash\xspace}
```

Here the `xspace` package comes in handy to avoid swallowed spaces.
For lists you may want to try the `easylist` package. Now writing a list is as simple as

```
\usepackage[ampersand]{easylist}
% ...

\begin{easylist}
& Item 1
& Item 2
&& Subitem 1
&&& Subsubitem 1
& Item 3
&& Subitem 1
\end{easylist}
```

58.21. Reducing the size of your LaTeX installation

The Installation[13] article explains in detail how to manually install a fully functional TeX environment, including LaTeX and other features, in under 100 MB.

9 Chapter 14 on page 153
10 Chapter 14 on page 153
11 Chapter 14 on page 153
12 Chapter 2.3 on page 17
13 Chapter 2 on page 11

59. Tips and Tricks

59.1. Always writing LaTeX in roman

If you insert the `\LaTeX` command in an area with a non-default font, it will be formatted accordingly. If you want to keep LaTeX written in Computer Modern roman shape, you must redefine the function. However, the naive

```
\renewcommand{\LaTeX}{{\rm \LaTeX}}
```

will output:

```
TeX capacity exceeded , sorry [ grouping levels =255].
```

So you need to create a temporary variable.
Sadly,

```
\newcommand{\LaTeXtemp}{\LaTeX}
\renewcommand{\LaTeX}{{\rm \LaTeXtemp}}
```

does not work as well.
We must use the TeX primitive `\let` instead.

```
\let\LaTeXtemp\LaTeX
\renewcommand{\LaTeX}{{\rm \LaTeXtemp }}
```

59.2. *id est* and *exempli gratia* (i.e. and e.g.)

If you simply use the forms "`i.e. `" or "`e.g. `", LaTeX will treat the periods as end of sentence periods (i.e. full stop[1]) since they are followed by a space, and add more space before the next "sentence". To prevent LaTeX from adding space after the last period, the correct syntax is either "`i.e.\ `" or "`e.g.\ `".
Depending on style (e.g., *The Chicago Manual of Style*[2]), a comma can be used afterwards, which is interpreted by LaTeX as part of a sentence, since the period is not followed by any space. In this case, "`i.e., `" and "`e.g., `" do not need any special attention.
If the command `\frenchspacing` is used in the preamble, the space between sentences is always consistent.

59.3. Grouping Figure/Equation Numbering by Section

For long documents the numbering can become cumbersome as the numbers reach into double and triple digits. To reset the counters at the start of each section and prefix the numbers by the section number, include the following in the preamble.

```
\usepackage{amsmath}
```

1 `http://en.wikipedia.org/wiki/Full%20stop`
2 `http://en.wikipedia.org/wiki/The%20Chicago%20Manual%20of%20Style`

```
\numberwithin{equation}{section}
\numberwithin{figure}{section}
```

The same can be done with similar counter types and document units such as ”subsection”.

59.4. Generic header

As explained in the previous sections, a LaTeX source can be used to generate both a DVI and a PDF file. For very basic documents the source is the same but, if the documents gets more complicated, it could be necessary to make some changes in the source so that it will work for a format but it will not for the other. For example, all that is related to graphics has to be adapted according to the final format. As discussed in the section about floating objects, even if you should use different pictures according to the final format, you can override this limit putting in the same folder pictures in different formats (e.g., EPS and PNG) with the same name and link them without writing the extension. There is a simple way to solve this problem:

```
\usepackage{ifpdf}
```

or, if you don’t have this package, you can add the following text just after `\documentclass[...]{...}` :

```
\newif\ifpdf
\ifx\pdfoutput\undefined
  \pdffalse
\else
  \ifnum\pdfoutput=1
    \pdftrue
  \else
    \pdffalse
  \fi
\fi
```

this is plain TeX code. The *ifpdf* package and this code, both define a new *if-else* you can use to change your code according to the compiler you are using. After you have used this code, you can use whenever you want in your document the following syntax:

```
\ifpdf
    % we are running pdflatex
\else
    % we are running latex
\fi
```

place after `\ifpdf` the code you want to insert if you are compiling with *pdflatex* , place after `\else` the code you want to insert if you are compiling with *latex* . For example, you can use this syntax to load different packages or different graphic file formats[3] according to the compiler.

59.5. Graphics and Graph editors

59.5.1. Vector image editors with LaTeX support

It is often preferable to use the same font and font size in your images as in the document. Moreover, for scientific images, you may need mathematical formulae or special characters (such as Greek letters). Both things can be achieved easily if the image editor allows you

3 Chapter 17.4 on page 212

to use LaTeX code in your image. Most vector image editors do not offer this option. There are, however, a few exceptions.
In early days, LaTeX users used Xfig[4] for their drawings. The editor is still used by quite a few people nowadays because it has special 'export to LaTeX' features. It also gives you some very basic ways of encapsulating LaTeX text and math in the image (setting the text's 'special flag' to 'special' instead of 'normal'). When exporting, all LaTeX text will be put in a .tex-file, separately from the rest of the image (which is put in a .ps file).
A newer and easier-to-use vector image editor specially tailored to LaTeX use is IPE[5]. It allows any LaTeX command, including but not limited to mathematical formulae in the image. The program saves its files as editable .eps or .pdf files, which eliminates the need of exporting your image each time you have edited it.
A very versatile vector image editor is Inkscape[6]. It does not support LaTeX text by itself, but you can use the plugin Textext[7] for that. This allows you to put any block of LaTeX code in your image. Additionally since version 0.48 you can export to vectorgraphics with texts separated in a .tex file. Using this way text is rendered by the latex compiler itself.
LaTeXDraw is a free and open source graphical PSTricks generator and editor. It allows you to draw basic geometric objects and save the result in a variety of formats including .jpg, .png, .eps, .bmp as well as .tex. In the last case the saved file contains PSTricks/LaTeX code only. Owing to that you can include any possible LaTeX code in the picture, since the file is rendered by your LaTeX environment directly.
Another way to generate vectorgraphics is using the Asymptote[8] language. It is a programming language which produces vector images in encapsulated postscript format and supports LaTeX syntax in any textlabels.

59.5.2. Graphs with gnuplot

A simple method to include graphs and charts in LaTeX documents is to create it within a common spreadsheet software (OpenOffice Calc or MS Office Excel etc.) and include it in the document as a cropped screenshot. However, this produces poor quality rasterized images. Calc also allows you to copy-paste the charts into OpenOffice Draw and save them as PDF files.
Using Microsoft Excel 2010, charts can be copied directly to Microsoft Expression Design 4, where they can be saved as PDF files. These PDF files can be included in LaTeX. This method produces high quality vectorized images.
An excellent method to render graphs is through **gnuplot[9]** , a free and versatile plotting software, that has a special output filter directly for exporting files to LaTeX. We assume, that the data is in a CSV file (comma separated text) in the first and third column. A simple gnuplot script to plot the data can look like this:

4 `http://en.wikipedia.org/wiki/Xfig`
5 `http://en.wikipedia.org/wiki/Ipe_%28program%29`
6 `http://en.wikipedia.org/wiki/Inkscape`
7 `http://pav.iki.fi/software/textext/`
8 `http://en.wikipedia.org/wiki/Asymptote_%28vector_graphics_language%29`
9 `http://en.wikipedia.org/wiki/gnuplot`

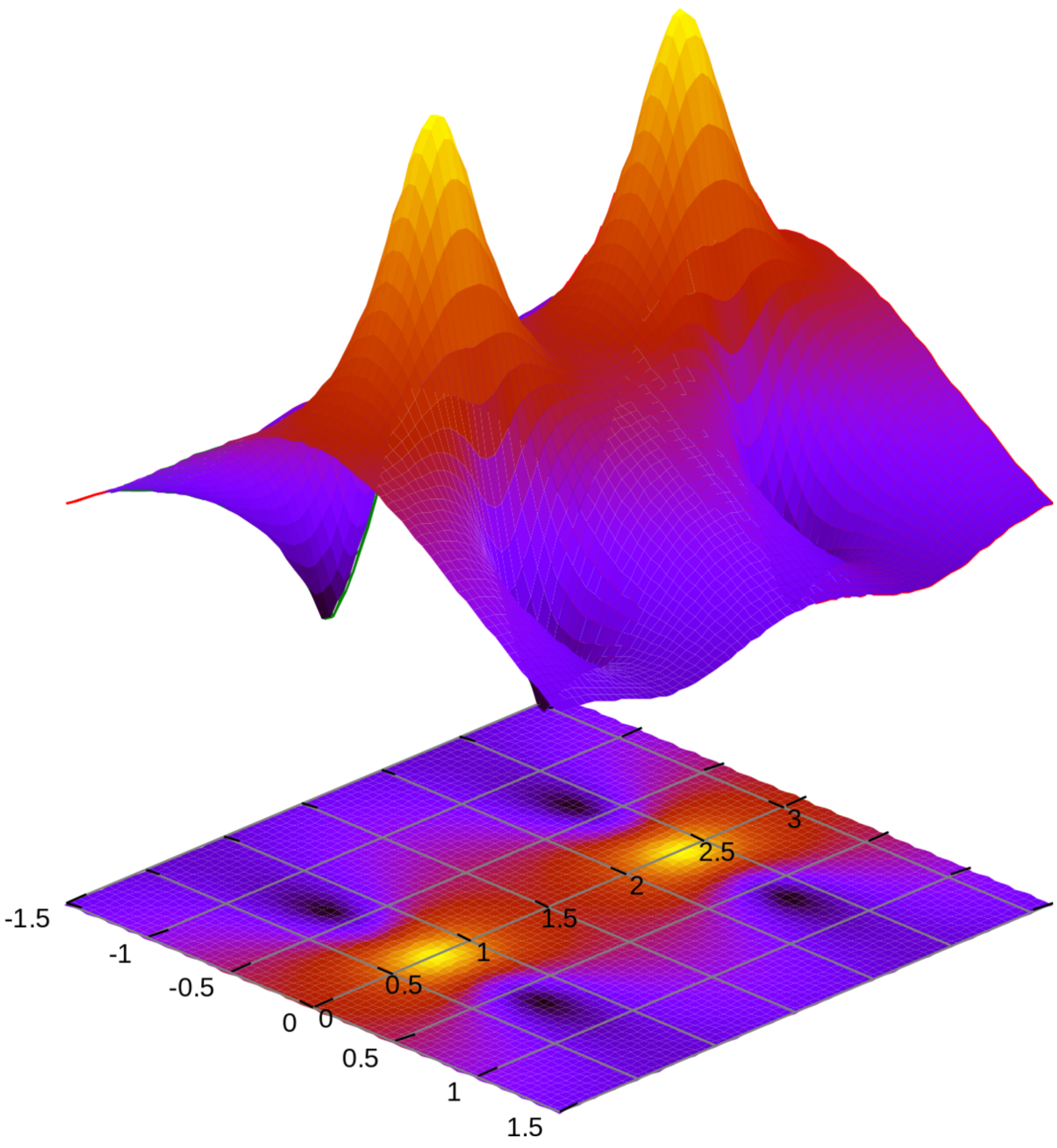


Figure 205 gnuplot can plot various numerical data, functions, error distribution as well as 3D graphs and surfaces

```
set format "$%g$"
set title "Graph 3: Dependence of $V_p$ on $R_0$"
set xlabel "Resistance $R_0$ [$\Omega$]"
set ylabel "Voltage $V_p$ [V]"
set border 3
set xtics nomirror
set ytics nomirror
set terminal epslatex
set output "graph1.eps"
plot "graph1.csv" using 1:3   #Plot the data
```

Now gnuplot produces two files: the graph drawing in `graph.eps` and the text in `graph.tex` . The second includes the EPS image, so that we only need to include the file graph.tex in our document:

```
\input{graph1.tex}
```

The above steps can be automated by the package gnuplottex. By placing gnuplot commands inside \begin{gnuplot}\end{gnuplot}, and compiling with latex -shell-escape, the graphs are created and added into your document.
Failure to access gnuplot from latex for Windows can be solved by making file title only in one word. Don't type **my report.tex** for your title file, but do **myreport.tex** .
When you are using gnuplottex it is also possible to directly pass the terminal settings as an argument to the environment

```
 \begin{gnuplot}[terminal=epslatex, terminaloptions=color, scale=0.9,
linewidth=2 ]
 ...
 \end{gnuplot}
```

Using gnuplottex can cause fraudulent text-highlighting in some editors when using algebraic functions on imported data, such as:

```
 (2*($1)):2
```

Some editors will think of all following text as part of a formula and highlight it as such (because of the '$' that is interpreted as part of the latex code). This can be avoided by ending with:

```
#$
\end{gnuplot}
```

As it uncomments the dollar sign for the gnuplot interpreter, but is not affecting the interpretation of the .tex by the editor.
When using pdfLaTeX instead of simple LaTeX, we must convert the EPS image to PDF and to substitute the name in the `graph1.tex` file. If we are working with a Unix-like shell, it is simply done using:

```
eps2pdf graph1.eps
sed -i s/".eps"/".pdf"/g graph1.tex
```

With the included tex file we can work as with an ordinary image.
Instead of calling `eps2pdf` directly, we can also include the `epstopdf` package that automates the process. If we include a graphics now and leave out the file extension, `epstopdf` will automatically transform the .eps-file to PDF and insert it in the text.

```
\includegraphics{graph1}
```

This way, if we choose to output to PS or DVI, the EPS version is used and if we output to PDF directly, the converted PDF graphics is used. Please note that usage of `epstopdf` requires compiling with latex -shell-escape.

Note: Emacs AucTex users might want to check out Gnuplot-mode[10].

59.5.3. Generate png screenshots

See Export To Other Formats[11].

59.6. Spell-checking and Word Counting

If you want to spell-check your document, you can use the command-line `aspell` , `hunspell` (preferably), or `ispell` programs.

```
ispell yourfile.tex
aspell --mode=tex -c yourfile.tex
hunspell -l -t -i utf-8 yourfile.tex
```

All three understand LaTeX and will skip LaTeX commands. You can also use a LaTeX editor with built-in spell checking, such as LyX[12], Kile[13], or Emacs[14]. Last another option is to convert LaTeX source to plain text[15] and open resulting file in a word processor like OpenOffice.org or KOffice.
If you want to count words you can, again, use LyX or convert your LaTeX source to plain text and use, for example, UNIX `wc` command:

```
detex yourfile | wc
```

An alternative to the `detex` command is the `pdftotext` command which extracts an ASCII text file from PDF:

```
1. pdflatex yourfile.tex
2. pdftotext yourfile.pdf
3. wc yourfile.txt
```

59.7. New even page

In the twoside-mode you have the ability to get a new odd-side page by:

`\cleardoublepage`

However, LaTeX doesn't give you the ability to get a new even-side page. The following method opens up this;
The following must be put in your document preamble:

```
\usepackage{ifthen}

\newcommand{\newevenside}{
```

10 `http://cars9.uchicago.edu/~ravel/software/gnuplot-mode.html`
11 Chapter 57 on page 627
12 `http://en.wikipedia.org/wiki/LyX`
13 `http://en.wikipedia.org/wiki/Kile`
14 `http://en.wikipedia.org/wiki/Emacs`
15 Chapter 57.8 on page 633

```
        \ifthenelse{\isodd{\thepage}}{\newpage}{
        \newpage
        \phantom{placeholder} % doesn't appear on page
        \thispagestyle{empty} % if want no header/footer
        \newpage
        }
}
```

To active the new even-side page, type the following where you want the new even-side:

```
\newevenside
```

If the given page is an odd-side page, the next new page is subsequently an even-side page, and LaTeX will do nothing more than a regular \newpage. However, if the given page is an even page, LaTeX will make a new (odd) page, put in a placeholder, and make another new (even) page. A crude but effective method.

59.8. Sidebar with information

If you want to put a sidebar with information like copyright and author, you might want to use the `eso-pic` package. Example:

```
\usepackage{eso-pic}
...
\AddToShipoutPicture{%
  \AtPageLowerLeft{%
    \rotatebox{90}{%
        \begin{minipage}{\paperheight}
          \centering\textcopyright~\today{} Humble me
        \end{minipage} %
      }
    } %
  }%
```

If you want it on one page only, use the starred version of the *AddToShipoutPicture* command at the page you want it. (`\AddToShipoutPicture*{...}`)

59.9. Hide auxiliary files

If you're using pdflatex you can create a folder in which all the output files will be stored, so your top directory looks cleaner.

```
pdflatex -output-directory tmp
```

Please note that the folder tmp should exist. However if you're using a Unix-based system you can do something like this:

```
alias pdflatex='mkdir -p tmp; pdflatex -output-directory tmp'
```

Or for vim modify your .vimrc:

```
" use pdflatex
let g:Tex_DefaultTargetFormat='pdf'
let g:Tex_MultipleCompileFormats='pdf,dvi'
let g:Tex_CompileRule_pdf = 'mkdir -p tmp; pdflatex -output-directory tmp
-interaction=nonstopmode $*; cp tmp/*.pdf .'
```

Part XI.

Appendices

60. Authors

60.1. Included books

The following books have been included in this wikibook (or we are working on it!), with permission of the author:

- Andy Roberts' Getting to grips with Latex[1].
- Not So Short Introduction to LaTex2e[2] by Tobias Oetiker, Hubert Partl and Irene Hyna. We have contacted the authors by email asking for permission: they allowed us to use their material, but they never edited directly this wikibook. That book is released under the GPL, that is not compatible with the GFDL used here in Wikibooks. Anyway, we have the permission of the authors to use their work. You can freely copy text from that guide to here. If you find text on both the original book and here on Wikibooks, then that text is double licensed under GPL and GFDL. For more information about Tobias Oetiker and Hubert Partl, their websites are `http://it.oetiker.ch/` and `http://homepage.boku.ac.at/partl/` respectively.
- LaTeX Primer[3] from the Indian TeX Users Group. Their document is released under the *GNU Free Documentation License* , the same as Wikibooks, so we can include parts of their document as we wish. In any case, we have contacted Indian TeX Users Group and they allowed us to do it.
- David Wilkins' Getting started with LaTeX[4]. The book is not released under any free license, but we have contacted the author asking him for the permission to use parts of his book on Wikibooks. He agreed: his work is still protected but you are allowed to copy the parts you want on this Wikibook. If you see text on both the original work and here, then that part (and only that part) is released under the terms of GFDL, like any other text here on Wikibooks.

In progress

- Peter Flynn's Formatting information, a beginner's guide to typesetting with LaTeX[5]. We have contacted him by email asking for permission to use his work. The original book is released under the *GNU Free Documentation License* , the same as Wikibooks. For more information, his personal website is `http://silmaril.ie/cgi-bin/blog`.

60.2. Wiki users

Major contributors to the book on Wikibooks are:

- Alessio Damato[6]

1 `http://www.andy-roberts.net/misc/latex/index.html`
2 `http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf`
3 `http://sarovar.org/projects/ltxprimer/`
4 `http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/`
5 `http://www.ctan.org/tex-archive/info/beginlatex/beginlatex-3.6.pdf`
6 `http://en.wikibooks.org/wiki/User%3AAlejo2083`

- Jtwdog[7]
- Pierre Neidhardt[8]

7 `http://en.wikibooks.org/wiki/User%3AJtwdog`
8 `http://en.wikibooks.org/wiki/User%3AAmbrevar`

61. Links

w:TeX[1] w:LaTeX[2]
Here are some other online resources available:

61.0.1. Community

- The TeX Users Group[3] Includes links to free versions of (La)TeX for many kinds of computers.
- UK-TUG[4] The UK TeX Users' Group
- TUGIndia[5] The Indian TeX Users Group
- [news:comp.text.tex comp.text.tex] Newsgroup for (La)TeX related questions
- CTAN[6] hundreds of add-on packages and programs

61.0.2. Tutorials/FAQs

- Tobias Oetiker's Not So Short Introduction to LaTex2e:
  `http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf` also at `http://web.archive.org/web/20010603070337/http://people.ee.ethz.ch/~oetiker/lshort/lshort.pdf`
- Vel's introduction to LaTeX: What is it, why should you use it, who should use it and how to get started:
  `http://www.vel.co.nz/vel.co.nz/Blog/Entries/2009/11/4_LaTeX_Document_Preparation_System.html`
- Peter Flynn's beginner's guide (formatting):
  `http://www.ctan.org/tex-archive/info/beginlatex/beginlatex-3.6.pdf`
- The AMS Short Math Guide for LaTeX, a concise summary of math formula typesetting features
  `http://www.ams.org/tex/amslatex.html`
- amsmath users guide (PDF) and related files:
  `http://www.ctan.org/tex-archive/macros/latex/required/amslatex/math/`
- LaTeX Primer from the Indian TeX Users Group:
  `http://sarovar.org/projects/ltxprimer/`
- LaTeX Primer
  `http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/`
- PSTricks--fancy graphics exploiting PDF capabilities
  `http://sarovar.org/projects/pstricks/`

1 `http://en.wikipedia.org/wiki/TeX`
2 `http://en.wikipedia.org/wiki/LaTeX`
3 `http://www.tug.org/`
4 `http://uk.tug.org/`
5 `http://www.tug.org.in/`
6 `http://www.ctan.org/`

- PDFScreen--create LaTeX PDF files that have navigation buttons used for presentations:
  `http://sarovar.org/projects/pdfscreen/`
- David Bausum's list of TeX primitives (these are the fundamental commands used in TeX):
  `http://www.tug.org/utilities/plain/cseq.html`
- Leslie Lamport's manual for the commands that are unique to LaTeX (commands not used in plain TeX):
  `http://www.tex.uniyar.ac.ru/doc/latex2e.pdf`
- The UK TeX FAQ List of questions and answers that are frequently posted at comp.text.tex
  `http://www.tex.ac.uk/faq`
- TeX on Mac OS X: Guide to using TeX and LaTeX on a Mac
  `http://www.rna.nl/tex.html`
- Text Processing using LaTeX
  `http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/`
- The (La)TeX encyclopaedia
  `http://tex.loria.fr/index.html`
- Hypertext Help with LaTeX
  `http://www.giss.nasa.gov/latex/`
- EpsLatex: a very comprehensive guide to images, figures and graphics
  `http://www.ctan.org/tex-archive/info/epslatex.pdf`
- The Comprehensive LaTeX Symbol List (in PDF)
  `http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf`
- Getting to Grips with LaTeX (HTML) Collection of Latex tutorials taking you from the very basics towards more advanced topics
  `http://www.andy-roberts.net/misc/latex/index.html`
- Chapter 8 (about typesetting mathematics) of the *LaTeX companion*
  `http://www.macrotex.net/texbooks/latexcomp-ch8.pdf`

61.0.3. Reference

- LaTeX Project Site[7]
- The Comprehensive TeX Archive Network[8] Latest (La)TeX-related packages and software
- TeX Directory Structure[9], used by many (La)TeX distributions
- Natural Math[10] converts natural language math formulas to LaTeX representation
- Obsolete packages and commands[11]
- Lamport's book *LaTeX: A Document Preparation System*

7 `http://www.latex-project.org/`
8 `http://www.ctan.org`
9 `http://www.tug.org/tds/`
10 `http://www.math.missouri.edu/~stephen/naturalmath/`
11 `http://www.ctan.org/tex-archive/info/l2tabu/english/l2tabuen.pdf`

61.0.4. Templates

- A resource for free high quality LaTeX templates for a variety of applications[12]
- LaTeX template for writing PhD thesis[13], 2007
- UCL computer department thesis template[14]
- UT thesis template[15], 2006
- A template that supports an easy conversion to *.odt (*.doc), *.pdf and *.html in one run[16], 2009

12 http://www.LaTeXTemplates.com
13 http://openwetware.org/wiki/LaTeX_template_for_PhD_thesis
14 http://www.cs.ucl.ac.uk/students/mphil_phd/resources_for_research_students/latex_for_research_thesis
15 http://www.cs.utexas.edu/users/jbednar/latex/
16 http://code.google.com/p/latex-template

62. Package Reference

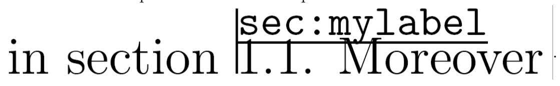
This is an incomplete list of useful packages that can be used for a wide range of different kind of documents. Each package has a short description next to it and, when available, there is a link to a section describing such package in detail. All of them (unless stated) should be included in your LaTeX distribution as *package_ name.sty* . For more information, refer to the documentation of the single packages, as described in Installing Extra Packages[1].
The list is in alphabetical order.

amsmath	It contains the advanced math extensions for LaTeX. The complete documentation should be in your LaTeX distribution; the file is called *amsdoc* , and can be *dvi* or *pdf* . For more information, see the chapter about Mathematics[2].
amssymb	It adds new symbols in to be used in math mode.
amsthm	It introduces the `proof` environment and the `\theoremstyle` command. For more information see the Theorems[3] section.
array	It extends the possibility of LaTeX to handle tables, fixing some bugs and adding new features. Using it, you can create very complicated and customized tables. For more information, see the Tables[4] section.
babel	It provides the internationalization of LaTeX. It has to be loaded in any document, and you have to give as an option the main language you are going to use in the document. For more information see the Internationalization[5] section.
bm	Allows use of bold greek letters in math mode using the `\bm{...}` command. This supersedes the `amsbsy` package.
booktabs	provides extra commands as well as behind-the-scenes optimisation for producing tables. Guidelines are given as to what constitutes a good table in the package documentation.
boxedminipage	It introduces the `boxedminipage` environment, that works exactly like `minipage` but adds a frame around it.
caption	Allows customization of appearance and placement of captions for figures, tables, etc.
cancel	Provides commands for striking out mathematical expressions. The syntax is`\cancel{x}`or`\cancelto{0}{x}`

changepage	To easily change the margins of pages. The syntax is
	`\changepage{textheight}{textwidth}%`
	`  {evensidemargin}{oddsidemargin}%`
	`  {columnsep}{topmargin}%`
	`  {headheight}{headsep}%`
	`  {footskip}`
	All the arguments can be both positive and negative numbers; they will be added (keeping the sign) to the relative variable.

cite	Supports compressed, sorted lists of numerical citations, and also deals with various punctuation and other issues of representation, including comprehensive management of break points.
color	It adds support for colored text. For more information, see the relevant section[6].
easylist	Adds support for arbitrarily-deep nested lists (useful for outlines). See List Structures[7].
esint	Adds additional integral symbols, for integrals over squares, clockwise integrals over sets, etc.
eucal	Other mathematical symbols.

1 Chapter 3 on page 29
2 Chapter 27 on page 305
3 Chapter 29 on page 367
4 Chapter 14 on page 153
5 Chapter 12 on page 133
6 Chapter 8 on page 89
7 Chapter 10 on page 109

fancyhdr	To change header and footer of any page of the document. It is described in the Page Layout[8] section.
fontenc	To choose the font encoding of the output text. You might need it if you are writing documents in a language other than English. Check in the Fonts[9] section.
geometry	For easy management of document margins and the document page size. See Page Layout[10].
glossaries	For creation of glossaries and list of acronyms. For more information, see the relevant chapter[11].
graphicx	allows you to insert graphic files within a document.
hyperref	It gives LaTeX the possibility to manage links within the document or to any URL when you compile in PDF. For more information, see the relevant section[12].
indentfirst	Once loaded, the beginning of any chapter/section is indented by the usual paragraph indentation.
inputenc	To choose the encoding of the input text. You might need it if you are writing documents in a language other than English. Check in the Special Characters[13] section.
latexsym	Other mathematical symbols.
listings	To insert programming code within the document. Many languages are supported and the output can be customized. For more information, see the Source Code Listings[14].
mathptmx	Sets the default font of the entire document (including math formulae) to Times New Roman, which is a more familiar font, and useful in saving space when fighting against page limits.
mathrsfs	Other mathematical symbols.
mhchem	allows you to easily type chemical species and equations. It automatically formats chemical species so you don't have to use subscript commands. It also Allows you to draw chemical formulas.
microtype	It provides an improvement to LaTeX's default typographic extensions, improvements in such areas as character protrusion and font expansion, interword spacing and additional kerning, and hyphenatable letter-spacing
multicol	provides the multicols environment which typesets text into multiple columns.
natbib	Gives additional citation options and styles.
paralist	provides compactitem environment which typesets list items much more closely than LaTeX's default.
pdfpages	This package simplifies the insertion of external multi-page PDF or PS documents.
rotating	It lets you rotate any kind of object. It is particularly useful for rotating tables. For more information, see the relevant section[15].
setspace	Lets you change line spacing, e.g. provides the `\doublespacing` command for making double spaced documents. For more information, see the relevant section[16].
showkeys	A useful package related to referencing. If you wish to reference an image or formula, you have to give it a name using `\label{...}` and then you can recall it using `\ref{...}`. When you compile the document these will be replaced only with numbers, and you can't know which label you had used unless you take a look at the source. If you have loaded the `showkeys` package, you will see the label just next or above the relevant number in the compiled version. An example of a reference to a section is  **Figure 206** This way you can easily keep track of the labels you add or use, simply looking at the preview (both *dvi* or *pdf*). Just before the final version, remove it.
showidx	It prints out all index entries in the left margin of the text. This is quite useful for proofreading a document and verifying the index. For more information, see the Indexing[17] section.
subfiles	The "root" and "child" document can be compiled at the same time without making changes to the "child" document. For more information, see the Modular Documents[18] section.
subcaption	It allows to define multiple floats (figures, tables) within one environment giving individual captions and labels in the form 1a, 1b.

8 Chapter 16 on page 193
9 Chapter 9 on page 95
10 Chapter 16 on page 193
11 Chapter 35 on page 429
12 Chapter 20 on page 255
13 Chapter 11 on page 123
14 Chapter 32 on page 393
15 Chapter 13 on page 151
16 Chapter 6 on page 65
17 Chapter 34 on page 421
18 Chapter 55 on page 607

syntonly	If you add the following code in your preamble:
	`\usepackage{syntonly}`
	`\syntaxonly`
	 LaTeX skims through your document only checking for proper syntax and usage of the commands, but doesn't produce any (DVI or PDF) output. As LaTeX runs faster in this mode you may save yourself valuable time. If you want to get the output, you can simply comment out the second line.

textcomp	Provides extra symbols, e.g. arrows like `\textrightarrow`, various currencies (`\texteuro`,...), things like `\textcelsius` and many others.
theorem	You can change the style of newly defined theorems. For more information see the Theorems[19] section.
todonotes	Lets you insert notes of stuff to do with the syntax `\todo{Add details.}`.
siunitx [20]	Helps you typeset of SI-units correctly. For example `\SI{12}{\mega\hertz}`. Automatically handles the correct spacing between the number and the unit. Note that even non-SI-units are set, like dB, rad, ...
ulem	It allows to underline text (either with straight or wavy line). Few examples of usage are added to the Fonts[21] chapter.
url	It defines the `\url{...}` command. URLs often contain special character such as '_' and '&', in order to write them you should *escape* them inserting a backslash, but if you write them as an argument of `\url{...}`, you don't need to escape any special character and it will take care of proper formatting for you. If you are using `hyperref`, you don't need to load `url` because it already provides the `\url{...}` command.
verbatim	It improves the `verbatim` environment, fixing some bugs. Moreover, it provides the `comment` environment, that lets you add multiple-line comments or easily comment out big parts of the code.
wrapfig	To insert images surrounded by text. It was discussed in section Floats, Figures and Captions[22].
xypic	It is used to create trees, graphs, (commutative) diagrams, and similar things. See Xy-pic[23].

19 Chapter 29 on page 367
20 `http://ctan.org/tex-archive/macros/latex/contrib/siunitx`
21 Chapter 9 on page 95
22 Chapter 18 on page 231
23 Chapter 49 on page 559

63. Sample LaTeX documents

The easiest way to learn how to use latex is to look at how other people use it. Here is a list of *real world* latex sources that are freely available on the internet. The information here is sorted by application area, so that it is grouped by the scientific communities that use similar notation and LaTeX constructs.

63.1. General examples

Tutorial examples, books, and real world uses of LaTeX.

- caption.tex[1], simple.tex[2], wrapped.tex[3]
- [ftp://tug.ctan.org/tex-archive/macros/latex/base/small2e.tex small2e.tex] and [ftp://tug.ctan.org/tex-archive/macros/latex/base/sample2e.tex sample2e.tex]. The "official" sample documents...
- A short example of how to use LaTeX for scientific reports[4] by Stephen J. Eglen.
- The not so Short Introduction to LaTeX[5] by Tobias Oetiker is distributed with full latex sources.

63.2. Semantics of Programming Languages

Articles on programming language research, from syntax to semantics, including source code listings, type rules, proof trees, and even some category theory. A good place to start is Mitchell Wand's Latex Resources[6], including a sample file that also demonstrates Didier Remy's mathpartir[7] package. The following are latex sources of some articles, books, or presentations from this field:

- Pugs: Bootstrapping Perl 6 with Haskell[8]. This paper by Audrey Tang contains nice examples on configuring the listings package[9] to format source code.

1 `http://en.wikibooks.org/wiki/LaTeX%2Fcaption.tex`
2 `http://en.wikibooks.org/wiki/LaTeX%2Fsimple.tex`
3 `http://en.wikibooks.org/wiki/LaTeX%2Fwrapped.tex`
4 `http://www.tug.org/pracjourn/2006-2/eglen/`
5 `http://www.ctan.org/tex-archive/info/lshort/english/`
6 `http://www.ccs.neu.edu/course/csg264/latex/`
7 `http://cristal.inria.fr/~remy/latex/`
8 `http://svn.openfoundry.org/pugs/docs/talks/hw2005.tex`
9 `http://en.wikibooks.org/wiki/LaTeX%2FPackages%2FListings`

64. Index

This is an alphabetical index of the book.

1 Chapter 4 on page 37
2 Chapter 7.6.5 on page 87
3 Chapter 7.6.1 on page 85
4 Chapter 31 on page 383
5 Chapter 27.10 on page 320
6 Chapter 60 on page 653
7 Chapter 12 on page 133
8 Chapter 4 on page 37
9 Chapter 41 on page 491
10 Chapter 38 on page 443
11 Chapter 38 on page 443
12 Chapter 7.6.1 on page 85
13 Chapter 10 on page 109
14 Chapter 10 on page 109
15 Chapter 18 on page 231
16 Chapter 56 on page 615

17 Chapter 8 on page 89
18 Chapter 8 on page 89
19 Chapter 16.9 on page 208
20 Chapter 21 on page 267
21 Chapter 51 on page 569
22 Chapter 7.6.1 on page 85
23 Chapter 7.6.1 on page 85
24 Chapter 7.6.1 on page 85
25 Chapter 4.5.6 on page 48
26 Chapter 5 on page 51
27 Chapter 44 on page 517
28 Chapter 59.2 on page 643
29 Chapter 7.6.1 on page 85
30 Chapter 7.6.1 on page 85
31 Chapter 7.6.1 on page 85
32 Chapter 7.6.1 on page 85
33 Chapter 22 on page 277
34 Chapter 7.6.1 on page 85
35 Chapter 57 on page 627
36 Chapter 18 on page 231

37 Chapter 18 on page 231
38 Chapter 9 on page 95
39 Chapter 16 on page 193
40 Chapter 7.6.1 on page 85
41 Chapter 7 on page 79
42 Chapter 55 on page 607
43 Chapter 44 on page 517
44 Chapter 17 on page 211
45 Chapter 17 on page 211
46 Chapter 17 on page 211
47 Chapter 16 on page 193
48 Chapter 57.6 on page 631
49 Chapter 20 on page 255
50 Chapter 20 on page 255
51 Chapter 7.6.1 on page 85
52 Chapter 7.6.1 on page 85
53 Chapter 59.2 on page 643
54 Chapter 17 on page 211
55 Chapter 17 on page 211

56 Chapter 34 on page 421
57 Chapter 12 on page 133
58 Chapter 1 on page 5
59 Chapter 7.6.1 on page 85
60 Chapter 7.6.1 on page 85
61 Chapter 21 on page 267
62 Chapter 40 on page 485
63 Chapter 61 on page 655
64 Chapter 10 on page 109
65 Chapter 34 on page 421
66 Chapter 5.3.1 on page 55
67 Chapter 7.6.1 on page 85
68 Chapter 44 on page 517
69 Chapter 27 on page 305
70 Chapter 27.10 on page 320
71 Chapter 15 on page 187
72 Chapter 16.9 on page 208
73 Chapter 62 on page 659
74 Chapter 51 on page 569
75 Chapter 16 on page 193

76 Chapter 57.3.4 on page 630
77 Chapter 44 on page 517
78 Chapter 17 on page 211
79 Chapter 57.3.4 on page 630
80 Chapter 41 on page 491
81 Chapter 31 on page 383
82 Chapter 7.6.4 on page 87
83 Chapter 21 on page 267
84 Chapter 57.3.4 on page 630
85 Chapter 6.1.3 on page 66
86 Chapter 7.6.1 on page 85
87 Chapter 7.6.1 on page 85
88 Chapter 7.6.1 on page 85
89 Chapter 59.6 on page 648
90 Chapter 27.5 on page 309
91 Chapter 6.10 on page 73
92 Chapter 57.3.4 on page 630
93 Chapter 5.3.5 on page 59

94 Chapter 14 on page 153
95 Chapter 7.6.1 on page 85
96 Chapter 15 on page 187
97 Chapter 29 on page 367
98 Chapter 59 on page 643
99 Chapter 15 on page 187
100 Chapter 7.6.1 on page 84
101 Chapter 7.6.1 on page 82
102 Chapter 59.6 on page 648
103 Chapter 9.10 on page 107
104 Chapter 49 on page 559
105 Chapter 49 on page 559

65. Command Glossary

This is a glossary of LaTeX commands—an alphabetical listing of LaTeX commands with the summaries of their effects. (Brackets ”[]” are optional arguments and braces ”{}” are required arguments.)

65.1.

/
 see slash marks[1]
@
 following period ends sentence
\\**[*][extra-space]**
 new line
,
 thin space, math and text mode
;
 thick space, math mode
:
 medium space, math mode
!
 negative thin space, math mode
-
 hyphenation; tabbing
\\=
 set tab, see tabbing
\\>
 tab, see tabbing
\\<
 back tab, see tabbing
\\+
 see tabbing
’
 accent or tabbing
‘
 accent or tabbing
\\|
 double vertical lines, math mode
(
 start math environment[2]

1 Chapter 7.6.1 on page 85
2 Chapter 27 on page 305

\)
end math environment
\[
begin displaymath environment
\]
end displaymath environment

65.2. A

\addcontentsline{file}{sec_unit}{entry}
adds an entry to the specified list or table
\addtocontents{file}{text}
adds text (or formatting commands) directly to the file that generates the specified list or table
\addtocounter{counter}{value}
increments the counter
\address{Return address}
\addtolength{len-cmd}{len}
increments a length command, see Length[3]
\addvspace
adds a vertical space of a specified height
\alph
causes the current value of a specified counter to be printed in alphabetic characters
\appendix
changes the way sectional units are numbered so that information after the command is considered part of the appendix
\arabic
causes the current value of a specified counter to be printed in Arabic numbers
\author
declares the author(s). See Document Structure[4]

65.3. B

\backslash
prints a backslash
\baselineskip
a length command (see Lengths[5]), which specifies the minimum space between the bottom of two successive lines in a paragraph
\baselinestretch
scales the value of \baselineskip
\bf
Boldface typeface
\bibitem

3 Chapter 23 on page 283
4 Chapter 5.3.1 on page 55
5 Chapter 23 on page 283

generates a labeled entry for the bibliography[6]

\bigskipamount

\bigskip

equivalent to \vspace{\bigskipamount}

\boldmath

bold font in math mode

\boldsymbol

bold font for symbols

65.4. C

\cal

Calligraphic style in math mode

\caption

generate caption for figures and tables

\cdots

Centered dots

\centering

Used to center align LaTeX environments

\chapter

Starts a new chapter. See Document Structure[7].

\circle

\cite

Used to make citations[8] from the provided bibliography

\cleardoublepage

\clearpage

Ends the current page and causes any floats to be printed. See Page Layout[9].

\cline

Adds horizontal line in a table that spans only to a range of cells. See \hline[10] and ../Tables/[11] chapter.

\closing

Inserts a closing phrase (e.g. \closing{yours sincerely}), leaves space for a handwritten signature and inserts a signature specified by \signature{}. Used in the *Letter* class.

\color

Specifies color of the text. ../Colors[12]

\copyright

makes © sign. See Formatting[13].

6 Chapter 5.3.1 on page 55
7 Chapter 5.3.3 on page 56
8 Chapter 38.2 on page 444
9 Chapter 16.10 on page 208
10 Chapter 65.9 on page 675
11 Chapter 14 on page 153
12 Chapter 8 on page 89
13 Chapter 7.6.1 on page 85

65.5. D

\dashbox
\date
\ddots
Inserts a diagonal ellipsis (3 diagonal dots) in math mode
\documentclass[options]{style}
Used to begin a latex document
\dotfill

65.6. E

\em
Toggles italics on/off for the text inside curly braces with the command. Such as {\em This is in italics \em but this isn't \em and this is again}. This command allows nesting.
\emph
Toggles italics on/off for the text in curly braces following the command e.g. \emph{This is in italics \emph{but this isn't} and this is again}.
\ensuremath (LaTeX2e)
Treats everything inside the curly braces as if it were in a math environment. Useful when creating commands in the preamble as they will work inside or out of math environments.
\epigraph
Adds an epigraph. Requires `epigraph` package.
\euro
Prints euro € symbol. Requires `eurosym` package.

65.7. F

\fbox
\flushbottom
\fnsymbol
\footnote
Creates a footnote[14].
\footnotemark
\footnotesize
Sets font size. See Text Formatting[15].
\footnotetext
\frac
inserts a fraction in mathematics mode. The usage is \frac{numerator}{denominator}.
\frame
\framebox
Like \makebox but creates a frame around the box. See Boxes[16].
\frenchspacing

14 Chapter 19.1 on page 249
15 Chapter 6.10 on page 73
16 Chapter 25 on page 295

Instructs LaTex to abstain from inserting more space after a period (`´.´`) than is the case for an ordinary character. In order to untoggle this functionality resort to the command \nonfrenchspacing[17].

65.8. G

65.9. H

\hfill
Abbreviation for \hspace{\fill}.
\hline
adds a horizontal line in a tabular environment. See also \cline[18], Tables[19] chapter.
\href
Add a link, or an anchor. See Hyperlinks[20]
\hrulefill
\hspace
Produces horizontal space.
\huge
Sets font size. See Text Formatting[21].
\Huge
Sets font size. See Text Formatting[22].
\hyphenation{word list}
Overrides default hyphenation algorithm for specified words. See Hyphenation[23]

65.10. I

\include
This command is different from `\input` in that it's the output that is added instead of the commands from the other files. For more see LaTex/Basics[24]
\includegraphics
Inserts an image[25]. Requires `graphicx` package.
\includeonly
\indent
\input
Used to read in LaTex files. For more see LaTex/Basics[26].
\it

17 Chapter 65.14 on page 677
18 Chapter 65.4 on page 673
19 Chapter 14 on page 153
20 Chapter 20 on page 255
21 Chapter 6.10 on page 73
22 Chapter 6.10 on page 73
23 Chapter 6.2 on page 67
24 Chapter 4.5.6 on page 48
25 Chapter 17.12.1 on page 226
26 Chapter 4.5.6 on page 48

Italicizes the text which is inside curly braces with the command. Such as {\it This is in italics}. \em is generally preferred since this allows nesting.

\item

Creates an item in a list. Used in list structures[27].

65.11. K

\kill

Prevent a line in the tabbing environment from being printed.

65.12. L

\label

Used to create label which can be later referenced with `\ref` . See Labels and Cross-referencing[28].

\large

Sets font size. See Text Formatting[29].

\Large

Sets font size. See Text Formatting[30].

\LARGE

Sets font size. See Text Formatting[31].

\LaTeX

Prints LaTeX logo. See Formatting[32].

\LaTeXe

Prints current LaTeX version logo. See Formatting[33].

\ldots

Prints sequence of three dots. See Formatting[34].

\left

\lefteqn

\line

\linebreak

Suggests LaTeX to break line in this place. See Page Layout[35].

\linethickness

\linewidth

\listoffigures

Inserts a list of the figures in the document. Similar to TOC[36]

\listoftables

27 Chapter 10 on page 109
28 Chapter 21 on page 267
29 Chapter 6.10 on page 73
30 Chapter 6.10 on page 73
31 Chapter 6.10 on page 73
32 Chapter 6.14 on page 78
33 Chapter 6.14 on page 78
34 Chapter 6.10 on page 73
35 Chapter 16.10 on page 208
36 Chapter 5.3.5 on page 59

Inserts a list of the tables in the document. Similar to TOC[37]

\location

65.13. M

\makebox

Defines a box that has a specified width, independent from its content. See Boxes[38].

\maketitle

Causes the title page to be typeset, using information provided by commands such as \title{} and \author{}.

\markboth \markright

\mathcal

\mathop

\mbox

Write a text in roman font inside a math part

\medskip

\multicolumn

\multiput

65.14. N

\newcommand

Defines a new command. See New Commands[39].

\newcolumntype

Defines a new type of column to be used with tables. See Tables[40].

\newcounter

\newenvironment

Defines a new environment. See New Environments[41].

\newfont

\newlength

\newline

Ends current line and starts a new one. See Page Layout[42].

\newpage

Ends current page and starts a new one. See Page Layout[43].

\newsavebox

\newtheorem

\nocite

Adds a reference to the bibliography without an inline citation. \nocite{*} causes all entries in a bibtex database to be added to the bibliography.

\noindent

37 Chapter 5.3.5 on page 59
38 Chapter 25 on page 295
39 Chapter 51.1 on page 569
40 Chapter 14 on page 153
41 Chapter 51.2 on page 571
42 Chapter 16.10 on page 208
43 Chapter 16.10 on page 208

\nolinebreak

\nonfrenchspacing

Setting the command untoggles the command \frenchspacing[44] and activates LaTeX standards to insert more space after a period (´.´) than after an ordinary character.

\normalsize

Sets default font size. See Text Formatting[45].

\nopagebreak

Suggests LaTeX not to break page in this place. See Page Layout[46].

\not

65.15. O

\onecolumn

\opening

Inserts an opening phrase when using the *letter* class, for example \opening{Dear Sir}

\oval

\overbrace

Draws a brace over the argument. Can be used in displaystyle with superscript to label formulae. See Advanced Mathematics[47].

\overline

Draws a line over the argument.

65.16. P

\pagebreak

Suggests LaTeX breaking page in this place. See Page Layout[48].

\pagenumbering

Defines the type of characters used for the page numbers. Options : arabic, roman, Roman, alph, Alph, gobble (invisible).

\pageref

Used to reference to number of page where a previously declared `\label` is located. See Floats, Figures and Captions[49].

\pagestyle

See Page Layout[50].

\par

Starts a new paragraph

\paragraph

Starts a new paragraph. See Document Structure[51].

\parbox

44 Chapter 65.7 on page 674
45 Chapter 6.10 on page 73
46 Chapter 16.10 on page 208
47 Chapter 28.2.1 on page 345
48 Chapter 16.10 on page 208
49 Chapter 18.5 on page 238
50 Chapter 16.7.1 on page 201
51 Chapter 5.3.3 on page 56

Defines a box whose contents are created in paragraph mode. See Boxes[52].

\parindent
Normal paragraph indentation. See Lengths[53].

\parskip

\part
Starts a new part of a book. See Document Structure[54].

\protect

\providecommand (LaTeX2e)
See Macros[55].

\put

65.17. Q

\quad
Similar to space, but with the size of a capital M

\qquad
double \quad

65.18. R

\raggedbottom
Command used for top justified within other environments.

\raggedleft
Command used for right justified within other environments.

\raggedright
Command used for left justified within other environments.

\raisebox
Creates a box and raises its content. See LaTeX/Boxes[56].

\ref
Used to reference to number of previously declared `\label` . See Labels and Cross-referencing[57].

\renewcommand

\right

\rm

\roman

\rule
Creates a line of specified width and height. See LaTeX/Rules and Struts[58].

52 Chapter 25 on page 295
53 Chapter 23 on page 283
54 Chapter 5.3.3 on page 56
55 Chapter 51.1 on page 569
56 Chapter 25 on page 295
57 Chapter 21 on page 267
58 Chapter 26 on page 301

65.19. S

\savebox
Makes a box and saves it in a named storage bin.
\sbox
The short form of \savebox with no optional arguments.
\sc
Small caps.
\scriptsize
Sets font size. See Text Formatting[59].
\section
Starts a new section. See Document Structure[60].
\setcounter
\setlength
\settowidth
\sf
Sans serif.
\shortstack
\signature
In the *Letter* class, specifies a signature for later insertion by \closing.
\sl
Slanted.
\slash
See slash marks[61]
\small
Sets font size. See Text Formatting[62].
\smallskip
\sout
Strikes out text. Requires `ulem` package. See Text Formatting[63].
\space
force ordinary space
\sqrt
Creates a root[64] (default square, but magnitude can be given as an optional parameter).
\stackrel
Takes two arguments and stacks the first on top of the second.
\stepcounter
Increase the counter.
\subparagraph
Starts a new subparagraph. See Document Structure[65].
\subsection

59 Chapter 6.10 on page 73
60 Chapter 5.3.3 on page 56
61 Chapter 7.6.1 on page 85
62 Chapter 6.10 on page 73
63 Chapter 6 on page 65
64 Chapter 27.7 on page 313
65 Chapter 5.3.3 on page 56

Starts a new subsection. See Document Structure[66].

\subsubsection

Starts a new sub-subsection. See Document Structure[67].

65.20. T

\tableofcontents

Inserts a table of contents (based on section headings) at the point where the command appears.

\telephone

In the *letter* class, specifies the sender's telephone number.

\TeX

Prints TeX logo. See Text Formatting[68].

\textbf{}

Sets bold font style. See Text Formatting[69].

\textcolor{}{}

Creates colored text. See Entering colored text[70].

\textit{}

Sets italic font style. See Text Formatting[71].

\textmd{}

Sets medium weight of a font. See Text Formatting[72].

\textnormal{}

Sets normal font. See Text Formatting[73].

\textrm{}

Sets roman font family. See Text Formatting[74].

\textsc{}

Sets font style to small caps. See Text Formatting[75].

\textsf{}

Sets sans serif font family. See Text Formatting[76].

\textsl{}

Sets slanted font style. See Text Formatting[77].

\texttt{}

Sets typewriter font family. See Text Formatting[78].

\textup{}

Sets upright shape of a font. See Text Formatting[79].

66 Chapter 5.3.3 on page 56
67 Chapter 5.3.3 on page 56
68 Chapter 6.14 on page 78
69 Chapter 6.10 on page 73
70 Chapter 8.2 on page 89
71 Chapter 6.10 on page 73
72 Chapter 6.10 on page 73
73 Chapter 6.10 on page 73
74 Chapter 6.10 on page 73
75 Chapter 6.10 on page 73
76 Chapter 6.10 on page 73
77 Chapter 6.10 on page 73
78 Chapter 6.10 on page 73
79 Chapter 6.10 on page 73

\textwidth
\textheight
\thanks
\thispagestyle
\tiny
Sets font size. See Text Formatting[80].
\title
\today
Writes current day. See Text Formatting[81].
\tt
\twocolumn
\typeout
\typein

65.21. U

\uline
Underlines text. Requires `ulem` package. See Formatting[82].
\underbrace
\underline
\unitlength
\usebox
\usecounter
\uwave
Creates wavy underline. Requires `ulem` package. See Formatting[83].

65.22. V

\value
\vbox{text}
Encloses a paragraph's text to prevent it from running over a page break
\vcenter
\vdots
Creates vertical dots. See Mathematics[84].
\vector
\verb
Creates inline verbatim text. See Formatting[85].
\vfill
\vline
\vphantom
\vspace

80 Chapter 6.10 on page 73
81 Chapter 6.14 on page 78
82 Chapter 7.6.1 on page 85
83 Chapter 7.6.1 on page 85
84 Chapter 27.17.2 on page 332
85 Chapter 7.6.1 on page 82

This page uses material from Dr. Sheldon Green's Hypertext Help with LaTeX.

66. Contributors

Edits	User
140	3mta3[1]
2	ABCD[2]
1	Adam majewski[3]
46	Adrignola[4]
1	AlanBarrett[5]
121	Alejo2083[6]
3	AllenZh[7]
998	Ambrevar[8]
5	Ans[9]
2	Anubhab91[10]
1	Arbitrarily0[11]
2	Arided[12]
25	Arnehe[13]
1	Arthurvogel[14]
1	Arunib[15]
1	Atallcostsky[16]
4	Atcovi[17]
1	Atiq ur Rehman[18]
2	Az1568[19]
4	Basenga[20]
2	Benjaminevans82˜enwikibooks[21]

1 http://en.wikibooks.org/wiki/User:3mta3
2 http://en.wikibooks.org/wiki/User:ABCD
3 http://en.wikibooks.org/wiki/User:Adam_majewski
4 http://en.wikibooks.org/wiki/User:Adrignola
5 http://en.wikibooks.org/wiki/User:AlanBarrett
6 http://en.wikibooks.org/wiki/User:Alejo2083
7 http://en.wikibooks.org/wiki/User:AllenZh
8 http://en.wikibooks.org/wiki/User:Ambrevar
9 http://en.wikibooks.org/wiki/User:Ans
10 http://en.wikibooks.org/wiki/User:Anubhab91
11 http://en.wikibooks.org/wiki/User:Arbitrarily0
12 http://en.wikibooks.org/wiki/User:Arided
13 http://en.wikibooks.org/wiki/User:Arnehe
14 http://en.wikibooks.org/wiki/User:Arthurvogel
15 http://en.wikibooks.org/wiki/User:Arunib
16 http://en.wikibooks.org/wiki/User:Atallcostsky
17 http://en.wikibooks.org/wiki/User:Atcovi
18 http://en.wikibooks.org/wiki/User:Atiq_ur_Rehman
19 http://en.wikibooks.org/wiki/User:Az1568
20 http://en.wikibooks.org/wiki/User:Basenga
21 http://en.wikibooks.org/wiki/User:Benjaminevans82%257Eenwikibooks

1	Benson Muite[22]
26	BiT[23]
1	Brammers[24]
8	Bumbulski[25]
1	Calimo[26]
1	CallumPoole[27]
2	CarsracBot[28]
5	Cerniagigante[29]
2	Chaojoker[30]
3	Chazz[31]
142	ChrisHodgesUK[32]
1	Chuckhoffmann[33]
1	Clebell[34]
4	CommonsDelinker[35]
2	Computermacgyver[36]
2	Conrad.Irwin[37]
1	Courcelles[38]
1	Crasshopper[39]
2	Cícero[40]
77	Dan Polansky[41]
1	Daniel Mietchen[42]
1	Darklama[43]
5	DavidMcKenzie[44]
102	Derbeth[45]
6	Dilaudid[46]

22 http://en.wikibooks.org/wiki/User:Benson_Muite
23 http://en.wikibooks.org/wiki/User:BiT
24 http://en.wikibooks.org/wiki/User:Brammers
25 http://en.wikibooks.org/wiki/User:Bumbulski
26 http://en.wikibooks.org/wiki/User:Calimo
27 http://en.wikibooks.org/wiki/User:CallumPoole
28 http://en.wikibooks.org/wiki/User:CarsracBot
29 http://en.wikibooks.org/wiki/User:Cerniagigante
30 http://en.wikibooks.org/wiki/User:Chaojoker
31 http://en.wikibooks.org/wiki/User:Chazz
32 http://en.wikibooks.org/wiki/User:ChrisHodgesUK
33 http://en.wikibooks.org/wiki/User:Chuckhoffmann
34 http://en.wikibooks.org/wiki/User:Clebell
35 http://en.wikibooks.org/wiki/User:CommonsDelinker
36 http://en.wikibooks.org/wiki/User:Computermacgyver
37 http://en.wikibooks.org/wiki/User:Conrad.Irwin
38 http://en.wikibooks.org/wiki/User:Courcelles
39 http://en.wikibooks.org/wiki/User:Crasshopper
40 http://en.wikibooks.org/wiki/User:C%25C3%25ADcero
41 http://en.wikibooks.org/wiki/User:Dan_Polansky
42 http://en.wikibooks.org/wiki/User:Daniel_Mietchen
43 http://en.wikibooks.org/wiki/User:Darklama
44 http://en.wikibooks.org/wiki/User:DavidMcKenzie
45 http://en.wikibooks.org/wiki/User:Derbeth
46 http://en.wikibooks.org/wiki/User:Dilaudid

143	Dirk Hünniger[47]
2	Dlituiev[48]
3	Dmb[49]
2	DmitriyZotikov[50]
3	Drewbie[51]
1	Edudobay[52]
2	Filip Dominec[53]
1	Flamenco108[54]
1	Fmccown[55]
2	Franklin Yu[56]
1	GPHemsley[57]
1	GavinMcGimpsey[58]
1	GorillaWarfare[59]
1	Greenbreen[60]
2	Gronau˜enwikibooks[61]
1	Gryllida[62]
1	Hagindaz[63]
2	Hahc21[64]
1	Hankwang[65]
1	Hannes Röst[66]
1	Harp[67]
4	He7d3r[68]
4	Henry Tallboys[69]
1	Herbythyme[70]
1	ILubeMyCucumbers20[71]

47 http://en.wikibooks.org/wiki/User:Dirk_H%25C3%25BCnniger
48 http://en.wikibooks.org/wiki/User:Dlituiev
49 http://en.wikibooks.org/wiki/User:Dmb
50 http://en.wikibooks.org/wiki/User:DmitriyZotikov
51 http://en.wikibooks.org/wiki/User:Drewbie
52 http://en.wikibooks.org/wiki/User:Edudobay
53 http://en.wikibooks.org/wiki/User:Filip_Dominec
54 http://en.wikibooks.org/wiki/User:Flamenco108
55 http://en.wikibooks.org/wiki/User:Fmccown
56 http://en.wikibooks.org/wiki/User:Franklin_Yu
57 http://en.wikibooks.org/wiki/User:GPHemsley
58 http://en.wikibooks.org/wiki/User:GavinMcGimpsey
59 http://en.wikibooks.org/wiki/User:GorillaWarfare
60 http://en.wikibooks.org/wiki/User:Greenbreen
61 http://en.wikibooks.org/wiki/User:Gronau%257Eenwikibooks
62 http://en.wikibooks.org/wiki/User:Gryllida
63 http://en.wikibooks.org/wiki/User:Hagindaz
64 http://en.wikibooks.org/wiki/User:Hahc21
65 http://en.wikibooks.org/wiki/User:Hankwang
66 http://en.wikibooks.org/wiki/User:Hannes_R%25C3%25B6st
67 http://en.wikibooks.org/wiki/User:Harp
68 http://en.wikibooks.org/wiki/User:He7d3r
69 http://en.wikibooks.org/wiki/User:Henry_Tallboys
70 http://en.wikibooks.org/wiki/User:Herbythyme
71 http://en.wikibooks.org/wiki/User:ILubeMyCucumbers20

1	Inductiveload[72]
1	Infinite0694[73]
1	IrfanAli[74]
23	Ish ishwar˜enwikibooks[75]
1	Itai[76]
4	JECompton˜enwikibooks[77]
1	JackPotte[78]
1	Jafeluv[79]
1	Jayk˜enwikibooks[80]
1	Jdgilbey[81]
3	Jevon[82]
2	Jguk[83]
1	Jianhui67[84]
51	Jimbotyson[85]
3	Jluttine[86]
1	Joaospam[87]
1	Jodi.a.schneider[88]
3	Joe Schmedley[89]
14	Jomegat[90]
30	Jonathan Webley[91]
2	JonnyJD[92]
1	Jotomicron[93]
5	Juliusross[94]
1	Jwchong[95]
1	Kayau[96]

72 http://en.wikibooks.org/wiki/User:Inductiveload
73 http://en.wikibooks.org/wiki/User:Infinite0694
74 http://en.wikibooks.org/wiki/User:IrfanAli
75 http://en.wikibooks.org/wiki/User:Ish_ishwar%257Eenwikibooks
76 http://en.wikibooks.org/wiki/User:Itai
77 http://en.wikibooks.org/wiki/User:JECompton%257Eenwikibooks
78 http://en.wikibooks.org/wiki/User:JackPotte
79 http://en.wikibooks.org/wiki/User:Jafeluv
80 http://en.wikibooks.org/wiki/User:Jayk%257Eenwikibooks
81 http://en.wikibooks.org/wiki/User:Jdgilbey
82 http://en.wikibooks.org/wiki/User:Jevon
83 http://en.wikibooks.org/wiki/User:Jguk
84 http://en.wikibooks.org/wiki/User:Jianhui67
85 http://en.wikibooks.org/wiki/User:Jimbotyson
86 http://en.wikibooks.org/wiki/User:Jluttine
87 http://en.wikibooks.org/wiki/User:Joaospam
88 http://en.wikibooks.org/wiki/User:Jodi.a.schneider
89 http://en.wikibooks.org/wiki/User:Joe_Schmedley
90 http://en.wikibooks.org/wiki/User:Jomegat
91 http://en.wikibooks.org/wiki/User:Jonathan_Webley
92 http://en.wikibooks.org/wiki/User:JonnyJD
93 http://en.wikibooks.org/wiki/User:Jotomicron
94 http://en.wikibooks.org/wiki/User:Juliusross
95 http://en.wikibooks.org/wiki/User:Jwchong
96 http://en.wikibooks.org/wiki/User:Kayau

7	Kazkaskazkasako[97]
1	Kenyon[98]
1	Kernigh[99]
2	Koavf[100]
27	Kri[101]
1	Krischik[102]
1	Krishnavedala[103]
3	Krisrose˜enwikibooks[104]
2	Kroolik[105]
1	LaTeX˜enwikibooks[106]
5	Lavaka[107]
1	Leyo[108]
1	LivingShadow[109]
2	LlamaAl[110]
1	MER-C[111]
1	Mabdul[112]
1	Marcus Cyron[113]
4	Marozols[114]
1	Martin von Wittich[115]
1	Matthias M.[116]
1	Matěj Grabovský[117]
1	Mckay[118]
19	Mcld[119]
4	Mecanismo[120]
1	MichaelBillington[121]

97 http://en.wikibooks.org/wiki/User:Kazkaskazkasako
98 http://en.wikibooks.org/wiki/User:Kenyon
99 http://en.wikibooks.org/wiki/User:Kernigh
100 http://en.wikibooks.org/wiki/User:Koavf
101 http://en.wikibooks.org/wiki/User:Kri
102 http://en.wikibooks.org/wiki/User:Krischik
103 http://en.wikibooks.org/wiki/User:Krishnavedala
104 http://en.wikibooks.org/wiki/User:Krisrose%257Eenwikibooks
105 http://en.wikibooks.org/wiki/User:Kroolik
106 http://en.wikibooks.org/wiki/User:LaTeX%257Eenwikibooks
107 http://en.wikibooks.org/wiki/User:Lavaka
108 http://en.wikibooks.org/wiki/User:Leyo
109 http://en.wikibooks.org/wiki/User:LivingShadow
110 http://en.wikibooks.org/wiki/User:LlamaAl
111 http://en.wikibooks.org/wiki/User:MER-C
112 http://en.wikibooks.org/wiki/User:Mabdul
113 http://en.wikibooks.org/wiki/User:Marcus_Cyron
114 http://en.wikibooks.org/wiki/User:Marozols
115 http://en.wikibooks.org/wiki/User:Martin_von_Wittich
116 http://en.wikibooks.org/wiki/User:Matthias_M.
117 http://en.wikibooks.org/wiki/User:Mat%25C4%259Bj_Grabovsk%25C3%25BD
118 http://en.wikibooks.org/wiki/User:Mckay
119 http://en.wikibooks.org/wiki/User:Mcld
120 http://en.wikibooks.org/wiki/User:Mecanismo
121 http://en.wikibooks.org/wiki/User:MichaelBillington

2	MichaelSchoenitzer[122]
1	Mike.lifeguard[123]
1	Modest Genius[124]
1	Mouselb[125]
1	Ms2ger[126]
81	Mwtoews[127]
2	NavarroJ[128]
1	Negative24[129]
2	Neoptolemus[130]
8	Nobelium[131]
1	Nux[132]
21	Orderud[133]
29	PAC[134]
32	PAC2[135]
1	Pamputt[136]
6	Panic2k4[137]
3	Paxinum[138]
112	Pi zero[139]
1	PiRSquared17[140]
11	Piksi[141]
1	Pmlineditor[142]
1	PsyberS[143]
3	QUBot[144]
26	QuiteUnusual[145]
5	Qwertyus[146]

122 http://en.wikibooks.org/wiki/User:MichaelSchoenitzer
123 http://en.wikibooks.org/wiki/User:Mike.lifeguard
124 http://en.wikibooks.org/wiki/User:Modest_Genius
125 http://en.wikibooks.org/wiki/User:Mouselb
126 http://en.wikibooks.org/wiki/User:Ms2ger
127 http://en.wikibooks.org/wiki/User:Mwtoews
128 http://en.wikibooks.org/wiki/User:NavarroJ
129 http://en.wikibooks.org/wiki/User:Negative24
130 http://en.wikibooks.org/wiki/User:Neoptolemus
131 http://en.wikibooks.org/wiki/User:Nobelium
132 http://en.wikibooks.org/wiki/User:Nux
133 http://en.wikibooks.org/wiki/User:Orderud
134 http://en.wikibooks.org/wiki/User:PAC
135 http://en.wikibooks.org/wiki/User:PAC2
136 http://en.wikibooks.org/wiki/User:Pamputt
137 http://en.wikibooks.org/wiki/User:Panic2k4
138 http://en.wikibooks.org/wiki/User:Paxinum
139 http://en.wikibooks.org/wiki/User:Pi_zero
140 http://en.wikibooks.org/wiki/User:PiRSquared17
141 http://en.wikibooks.org/wiki/User:Piksi
142 http://en.wikibooks.org/wiki/User:Pmlineditor
143 http://en.wikibooks.org/wiki/User:PsyberS
144 http://en.wikibooks.org/wiki/User:QUBot
145 http://en.wikibooks.org/wiki/User:QuiteUnusual
146 http://en.wikibooks.org/wiki/User:Qwertyus

1	RainCity471[147]
3	Ramac[148]
1	Raylu[149]
1	RaymondSutanto[150]
10	Recent Runes[151]
6	Reddraggone9[152]
1	Redirect fixer[153]
1	Reyk[154]
1	Ricordisamoa[155]
1	Risk[156]
1	Rnddim[157]
84	Robbiemorrison[158]
1	Robert Borkowski[159]
4	Robert Horning[160]
3	Robin˜enwikibooks[161]
2	Sandbergja[162]
4	Sanderd17[163]
1	Sandrobt[164]
1	SciYann[165]
1	Scruss[166]
1	Simeon[167]
2	Sjlegg[168]
4	Spirosdenaxas[169]
1	Staticshakedown[170]
3	Stephan Schneider[171]

147 http://en.wikibooks.org/wiki/User:RainCity471
148 http://en.wikibooks.org/wiki/User:Ramac
149 http://en.wikibooks.org/wiki/User:Raylu
150 http://en.wikibooks.org/wiki/User:RaymondSutanto
151 http://en.wikibooks.org/wiki/User:Recent_Runes
152 http://en.wikibooks.org/wiki/User:Reddraggone9
153 http://en.wikibooks.org/wiki/User:Redirect_fixer
154 http://en.wikibooks.org/wiki/User:Reyk
155 http://en.wikibooks.org/wiki/User:Ricordisamoa
156 http://en.wikibooks.org/wiki/User:Risk
157 http://en.wikibooks.org/wiki/User:Rnddim
158 http://en.wikibooks.org/wiki/User:Robbiemorrison
159 http://en.wikibooks.org/wiki/User:Robert_Borkowski
160 http://en.wikibooks.org/wiki/User:Robert_Horning
161 http://en.wikibooks.org/wiki/User:Robin%257Eenwikibooks
162 http://en.wikibooks.org/wiki/User:Sandbergja
163 http://en.wikibooks.org/wiki/User:Sanderd17
164 http://en.wikibooks.org/wiki/User:Sandrobt
165 http://en.wikibooks.org/wiki/User:SciYann
166 http://en.wikibooks.org/wiki/User:Scruss
167 http://en.wikibooks.org/wiki/User:Simeon
168 http://en.wikibooks.org/wiki/User:Sjlegg
169 http://en.wikibooks.org/wiki/User:Spirosdenaxas
170 http://en.wikibooks.org/wiki/User:Staticshakedown
171 http://en.wikibooks.org/wiki/User:Stephan_Schneider

1	Swift[172]
1	Syum90[173]
1	TWiStErRob[174]
1	Tauriel-1[175]
1	Tdomhan[176]
1	Teles[177]
21	Thenub314[178]
1	Tim Parenti[179]
1	Tom Morris[180]
120	Tomato86[181]
3	Topodelapradera[182]
14	Tosha[183]
2	Towsonu2003~enwikibooks[184]
1	Trace[185]
1	Tualha[186]
1	Tweenk[187]
3	Urhixidur[188]
6	Vadik wiki[189]
77	Waldir[190]
1	Waylesange[191]
1	Whiteknight[192]
6	Whym[193]
14	Wickedjargon[194]
9	Wikieditoroftoday[195]
1	Winfree[196]

172 http://en.wikibooks.org/wiki/User:Swift
173 http://en.wikibooks.org/wiki/User:Syum90
174 http://en.wikibooks.org/wiki/User:TWiStErRob
175 http://en.wikibooks.org/wiki/User:Tauriel-1
176 http://en.wikibooks.org/wiki/User:Tdomhan
177 http://en.wikibooks.org/wiki/User:Teles
178 http://en.wikibooks.org/wiki/User:Thenub314
179 http://en.wikibooks.org/wiki/User:Tim_Parenti
180 http://en.wikibooks.org/wiki/User:Tom_Morris
181 http://en.wikibooks.org/wiki/User:Tomato86
182 http://en.wikibooks.org/wiki/User:Topodelapradera
183 http://en.wikibooks.org/wiki/User:Tosha
184 http://en.wikibooks.org/wiki/User:Towsonu2003%257Eenwikibooks
185 http://en.wikibooks.org/wiki/User:Trace
186 http://en.wikibooks.org/wiki/User:Tualha
187 http://en.wikibooks.org/wiki/User:Tweenk
188 http://en.wikibooks.org/wiki/User:Urhixidur
189 http://en.wikibooks.org/wiki/User:Vadik_wiki
190 http://en.wikibooks.org/wiki/User:Waldir
191 http://en.wikibooks.org/wiki/User:Waylesange
192 http://en.wikibooks.org/wiki/User:Whiteknight
193 http://en.wikibooks.org/wiki/User:Whym
194 http://en.wikibooks.org/wiki/User:Wickedjargon
195 http://en.wikibooks.org/wiki/User:Wikieditoroftoday
196 http://en.wikibooks.org/wiki/User:Winfree

25	Withinfocus[197]
12	Xania[198]
13	Xonqnopp[199]
1	Ynhockey[200]
1	Ysangkok[201]
3	ZeroOne[202]
3	Zvika[203]
1	bahn[204]
1	Ævar Arnfjörð Bjarmason[205]

197 http://en.wikibooks.org/wiki/User:Withinfocus
198 http://en.wikibooks.org/wiki/User:Xania
199 http://en.wikibooks.org/wiki/User:Xonqnopp
200 http://en.wikibooks.org/wiki/User:Ynhockey
201 http://en.wikibooks.org/wiki/User:Ysangkok
202 http://en.wikibooks.org/wiki/User:ZeroOne
203 http://en.wikibooks.org/wiki/User:Zvika
204 http://en.wikibooks.org/wiki/User:%2526bahn
205 http://en.wikibooks.org/wiki/User:%25C3%2586var_Arnfj%25C3%25B6r%25C3%25B0_Bjarmason

List of Figures

- GFDL: Gnu Free Documentation License. http://www.gnu.org/licenses/fdl.html
- cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License. http://creativecommons.org/licenses/by-sa/3.0/
- cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License. http://creativecommons.org/licenses/by-sa/2.5/
- cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License. http://creativecommons.org/licenses/by-sa/2.0/
- cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License. http://creativecommons.org/licenses/by-sa/1.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/deed.en
- cc-by-2.5: Creative Commons Attribution 2.5 License. http://creativecommons.org/licenses/by/2.5/deed.en
- cc-by-3.0: Creative Commons Attribution 3.0 License. http://creativecommons.org/licenses/by/3.0/deed.en
- GPL: GNU General Public License. http://www.gnu.org/licenses/gpl-2.0.txt
- LGPL: GNU Lesser General Public License. http://www.gnu.org/licenses/lgpl.html
- PD: This image is in the public domain.
- ATTR: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- EURO: This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- LFK: Lizenz Freie Kunst. http://artlibre.org/licence/lal/de
- CFR: Copyright free use.

- EPL: Eclipse Public License. `http://www.eclipse.org/org/documents/epl-v10.php`

Copies of the GPL, the LGPL as well as a GFDL are included in chapter Licenses[206]. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrower.

206 Chapter 67 on page 707

1	Gummi team. Original uploader was Sanderd17[207] at en.wikipedia[208]	
2	LyX developer team (see www.lyx.org)	GPL
3	PAC2[209]	GPL
4	BotMultichill, BotMultichillT, Emijrpbot, Hazard-Bot, JarektBot, KAMiKAZOW, Paucabot, Wiso	
5	Emijrpbot, Hazard-Bot, JarektBot, MGA73bot2, Mwtoews, Patrick87, SieBot, Ö	
6	Myself	
7	Alessio Damato	GFDL
8	Editor at Large, Infrogmation, Itsmine, JarektBot, Michiel1972, Shyam, Waldir, Wst	
9		
10		
11	Tomato86[210]	GFDL
12	Karl Scheel[211]	CC-BY-SA-3.0
13	Karl Scheel[212]	CC-BY-SA-3.0
14	Thenub314[213]	GFDL
15	Tomato86[214]	GFDL
16	Derbeth	
17	Tobias Oetiker	GFDL
18	Mike.lifeguard, Mwtoews	
19	Tobias Oetiker	GFDL
20	Apteva[215]	PD
21	Tobias Oetiker	GFDL
22	Tobias Oetiker	GFDL
23	Dirk Hünniger[216]	CC-BY-SA-3.0
24	Dirk Hünniger[217]	CC-BY-SA-3.0
25	ChrisHodgesUK[218]	PD
26	ChrisHodgesUK[219]	PD
27		
28	Dirk Hünniger[220]	CC-BY-SA-3.0
29	Dirk Hünniger[221]	CC-BY-SA-3.0
30	Dirk Hünniger[222]	CC-BY-SA-3.0

207 http://en.wikipedia.org/wiki/User:Sanderd17
208 http://en.wikipedia.org
209 http://commons.wikimedia.org/wiki/User:PAC2
210 http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1
211 http://commons.wikimedia.org/w/index.php?title=User:Kscheel&action=edit&redlink=1
212 http://commons.wikimedia.org/w/index.php?title=User:Kscheel&action=edit&redlink=1
213 http://commons.wikimedia.org/w/index.php?title=User:Thenub314&action=edit&redlink=1
214 http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1
215 http://commons.wikimedia.org/wiki/User:Apteva
216 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
217 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
218 http://commons.wikimedia.org/wiki/User:ChrisHodgesUK
219 http://commons.wikimedia.org/wiki/User:ChrisHodgesUK
220 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
221 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
222 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger

223 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
224 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
225 http://commons.wikimedia.org/w/index.php?title=User:Lanoxx&action=edit&redlink=1
226 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
227 http://commons.wikimedia.org/w/index.php?title=User:Arthurchy&action=edit&redlink=1
228 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
229 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
230 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
231 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
232 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
233 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
234 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
235 http://commons.wikimedia.org/wiki/User:ChrisHodgesUK
236 http://commons.wikimedia.org/wiki/User:ChrisHodgesUK
237 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
238 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
239 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
240 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
241 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
242 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
243 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
244 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
245 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
246 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
247 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger

57	Dirk Hünniger[248]	CC-BY-SA-3.0
58	Dirk Hünniger[249]	CC-BY-SA-3.0
59	Dirk Hünniger[250]	CC-BY-SA-3.0
60	Dirk Hünniger[251]	CC-BY-SA-3.0
61	Alessio Damato[252]	GFDL
62	The original uploader was Jtwdog[253] at English Wikibooks[254]	GFDL
63	The original uploader was Jtwdog[255] at English Wikibooks[256]	GFDL
64	The original uploader was Jtwdog[257] at English Wikibooks[258]	GFDL
65	ChrisHodgesUK[259]	PD
66	Alessio Damato[260]	GFDL
67	User:Mwtoews[261]	GFDL
68	Mwtoews[262]	GFDL
69	Alessio Damato[263]	GFDL
70	Alessio Damato[264]	GFDL
71	Alessio Damato[265]	GFDL
72	Derbeth	
73	Derbeth[266]	GFDL
74	Maschen[267]	PD
75	Alessio Damato[268]	GFDL
76	Alessio Damato[269] 13:31, 12 January 2007 (UTC)	GFDL
77	Alessio Damato[270]	GFDL
78	Ambrevar[271]	PD
79	Alessio Damato[272]	GFDL

248 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
249 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
250 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
251 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
252 http://commons.wikimedia.org/wiki/User:Alejo2083
253 http://en.wikibooks.org/wiki/User:Jtwdog
254 http://en.wikibooks.org/wiki/
255 http://en.wikibooks.org/wiki/User:Jtwdog
256 http://en.wikibooks.org/wiki/
257 http://en.wikibooks.org/wiki/User:Jtwdog
258 http://en.wikibooks.org/wiki/
259 http://commons.wikimedia.org/wiki/User:ChrisHodgesUK
260 http://commons.wikimedia.org/wiki/User:Alejo2083
261 http://commons.wikimedia.org/wiki/User:Mwtoews
262 http://commons.wikimedia.org/wiki/User:Mwtoews
263 http://commons.wikimedia.org/wiki/User:Alejo2083
264 http://commons.wikimedia.org/wiki/User:Alejo2083
265 http://commons.wikimedia.org/wiki/User:Alejo2083
266 http://commons.wikimedia.org/wiki/User:Derbeth
267 http://commons.wikimedia.org/wiki/User:Maschen
268 http://commons.wikimedia.org/wiki/User:Alejo2083
269 http://commons.wikimedia.org/wiki/User:Alejo2083
270 http://commons.wikimedia.org/wiki/User:Alejo2083
271 http://commons.wikimedia.org/wiki/User:Ambrevar
272 http://commons.wikimedia.org/wiki/User:Alejo2083

273 `http://commons.wikimedia.org/wiki/User:Alejo2083`
274 `http://commons.wikimedia.org/wiki/User:Alejo2083`
275 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
276 `http://commons.wikimedia.org/wiki/User:Alejo2083`
277 `http://commons.wikimedia.org/wiki/User:ChrisHodgesUK`
278 `http://commons.wikimedia.org/wiki/User:ChrisHodgesUK`
279 `http://commons.wikimedia.org/w/index.php?title=User:Winfree&action=edit&redlink=1`
280 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
281 `http://commons.wikimedia.org/wiki/User:Waldir`
282 `http://commons.wikimedia.org/wiki/User:Neet`
283 `http://commons.wikimedia.org/wiki/User:Neet`
284 `http://commons.wikimedia.org/wiki/User:Neet`
285 `http://commons.wikimedia.org/wiki/User:Neet`
286 `http://commons.wikimedia.org/wiki/User:Neet`
287 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
288 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
289 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
290 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
291 `http://commons.wikimedia.org/wiki/User:ChrisHodgesUK`
292 `http://commons.wikimedia.org/wiki/User:Inductiveload`
293 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
294 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
295 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
296 `http://commons.wikimedia.org/wiki/User:ChrisHodgesUK`
297 `http://commons.wikimedia.org/w/index.php?title=User:Tomato86&action=edit&redlink=1`
298 `http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1`

106	Pmillerrhodes[299]	CC-BY-SA-3.0
107	Pmillerrhodes[300]	CC-BY-SA-3.0
108	Ben Mills	
109	Pmillerrhodes[301]	CC-BY-SA-3.0
110	Pmillerrhodes[302]	CC-BY-SA-3.0
111	Pmillerrhodes[303]	CC-BY-SA-3.0
112	Pmillerrhodes[304]	CC-BY-SA-3.0
113	Pmillerrhodes[305]	CC-BY-SA-3.0
114	Daviewales[306]	
115	Daviewales[307]	
116	Pmillerrhodes[308]	CC-BY-SA-3.0
117	Pmillerrhodes[309]	CC-BY-SA-3.0
118	CategorizationBot, Edgar181, Jahobr, JarektBot, YaCBot	
119	Original uploader was Iorsh[310] at en.wikipedia[311]	PD
120	Lavaka[312]	CC-BY-SA-3.0
121	Nemti[313]	
122	MyName (Gkc[314] (talk[315]))	GFDL
123	LaTeX / GIMP	CC-BY-3.0
124	Ambrevar[316]	CC-BY-SA-3.0
125	jeg	PD
126	Hankjones[317]	GFDL
127	Hankjones[318]	CC-BY-SA-3.0
128	Hankjones[319]	CC-BY-SA-3.0
129	jeg	PD
130	Hankjones[320]	GFDL

299 http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1
300 http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1
301 http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1
302 http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1
303 http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1
304 http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1
305 http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1
306 http://commons.wikimedia.org/w/index.php?title=User:Daviewales&action=edit&redlink=1
307 http://commons.wikimedia.org/w/index.php?title=User:Daviewales&action=edit&redlink=1
308 http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1
309 http://commons.wikimedia.org/w/index.php?title=User:Pmillerrhodes&action=edit&redlink=1
310 http://en.wikipedia.org/wiki/User:Iorsh
311 http://en.wikipedia.org
312 http://commons.wikimedia.org/w/index.php?title=User:Lavaka&action=edit&redlink=1
313 http://commons.wikimedia.org/wiki/User:Nemti
314 http://commons.wikimedia.org/wiki/User:Gkc
315 http://commons.wikimedia.org/wiki/User_talk:Gkc
316 http://commons.wikimedia.org/wiki/User:Ambrevar
317 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
318 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
319 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
320 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1

321 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
322 http://commons.wikimedia.org/w/index.php?title=User:PhilJohnG&action=edit&redlink=1
323 http://commons.wikimedia.org/w/index.php?title=User:PhilJohnG&action=edit&redlink=1
324 http://commons.wikimedia.org/w/index.php?title=User:PhilJohnG&action=edit&redlink=1
325 http://commons.wikimedia.org/w/index.php?title=User:Matej.korvas&action=edit&redlink=1
326 http://commons.wikimedia.org/w/index.php?title=User:Matej.korvas&action=edit&redlink=1
327 http://commons.wikimedia.org/w/index.php?title=User:Olesh&action=edit&redlink=1
328 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
329 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
330 http://commons.wikimedia.org/wiki/User:ChrisHodgesUK
331 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
332 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
333 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
334 http://commons.wikimedia.org/w/index.php?title=User:Hankjones&action=edit&redlink=1
335 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
336 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
337 http://commons.wikimedia.org/wiki/User:Dirk_H%C3%BCnniger
338 http://commons.wikimedia.org/w/index.php?title=User:Literaturgenerator&action=edit&redlink=1
339 http://commons.wikimedia.org/w/index.php?title=User:Jimbotyson&action=edit&redlink=1
340 http://commons.wikimedia.org/w/index.php?title=User:Neoriddle&action=edit&redlink=1
341 http://commons.wikimedia.org/w/index.php?title=User:Flip&action=edit&redlink=1

159	Israel Buitron[342]	CC-BY-SA-3.0
160	Ambrevar[343]	PD
161	Ambrevar[344]	PD
162	ChrisHodgesUK[345]	PD
163	Alessio Damato[346]	GFDL
164	Alessio Damato[347]	GFDL
165	Alessio Damato[348]	GFDL
166	Alessio Damato[349]	GFDL
167	Alessio Damato[350]	GFDL
168	Alessio Damato[351]	GFDL
169	Alessio Damato[352]	GFDL
170	Alessio Damato[353]	GFDL
171	Alessio Damato[354]	GFDL
172	Alessio Damato[355]	GFDL
173	• Neighbourhood_definition2.png[356]: Wegmann[357] • derivative work: Pablo Castellanos[358] (talk[359])	GFDL
174	Nobelium[360]	PD
175	Nobelium[361]	PD
176	Nobelium[362]	PD
177	Nobelium[363]	PD
178	Nobelium[364]	PD
179	Nobelium[365]	
180	Nobelium[366]	
181	Nobelium[367]	

342 http://commons.wikimedia.org/w/index.php?title=User:Neoriddle&action=edit&redlink=1
343 http://commons.wikimedia.org/wiki/User:Ambrevar
344 http://commons.wikimedia.org/wiki/User:Ambrevar
345 http://commons.wikimedia.org/wiki/User:ChrisHodgesUK
346 http://commons.wikimedia.org/wiki/User:Alejo2083
347 http://commons.wikimedia.org/wiki/User:Alejo2083
348 http://commons.wikimedia.org/wiki/User:Alejo2083
349 http://commons.wikimedia.org/wiki/User:Alejo2083
350 http://commons.wikimedia.org/wiki/User:Alejo2083
351 http://commons.wikimedia.org/wiki/User:Alejo2083
352 http://commons.wikimedia.org/wiki/User:Alejo2083
353 http://commons.wikimedia.org/wiki/User:Alejo2083
354 http://commons.wikimedia.org/wiki/User:Alejo2083
355 http://commons.wikimedia.org/wiki/User:Alejo2083
356 http://commons.wikimedia.org/wiki/File:Neighbourhood_definition2.png
357 http://commons.wikimedia.org/wiki/User:Wegmann
358 http://commons.wikimedia.org/w/index.php?title=User:Pablo_Castellanos&action=edit&redlink=1
359 http://commons.wikimedia.org/wiki/User_talk:Pablo_Castellanos
360 http://commons.wikimedia.org/wiki/User:Nobelium
361 http://commons.wikimedia.org/wiki/User:Nobelium
362 http://commons.wikimedia.org/wiki/User:Nobelium
363 http://commons.wikimedia.org/wiki/User:Nobelium
364 http://commons.wikimedia.org/wiki/User:Nobelium
365 http://commons.wikimedia.org/wiki/User:Nobelium
366 http://commons.wikimedia.org/wiki/User:Nobelium
367 http://commons.wikimedia.org/wiki/User:Nobelium

182	Nobelium[368]	
183	Nobelium[369]	
184	Nobelium[370]	
185	Nobelium[371]	
186	Nobelium[372]	
187	Alessio Damato[373]	GFDL
188	Alessio Damato[374]	GFDL
189	Alessio Damato[375]	GFDL
190	Alessio Damato[376]	GFDL
191	Alessio Damato[377]	GFDL
192	Alessio Damato[378]	GFDL
193	Alessio Damato[379]	GFDL
194	Alessio Damato[380]	GFDL
195	Alessio Damato[381]	GFDL
196	Alessio Damato[382]	GFDL
197	Ambrevar[383]	PD
198	Ambrevar[384]	PD
199	Ambrevar[385]	PD
200	Original uploader was Arnehe[386] at en.wikibooks[387]	GPL
201	Original uploader was Arnehe[388] at en.wikibooks[389]	GPL
202	Original uploader was Arnehe[390] at en.wikibooks[391]	GPL
203	Original uploader was Arnehe[392] at en.wikibooks[393]	GPL
204	Original uploader was Arnehe[394] at en.wikibooks[395]	GPL
205	Alessio Damato[396]	GFDL

368 http://commons.wikimedia.org/wiki/User:Nobelium
369 http://commons.wikimedia.org/wiki/User:Nobelium
370 http://commons.wikimedia.org/wiki/User:Nobelium
371 http://commons.wikimedia.org/wiki/User:Nobelium
372 http://commons.wikimedia.org/wiki/User:Nobelium
373 http://commons.wikimedia.org/wiki/User:Alejo2083
374 http://commons.wikimedia.org/wiki/User:Alejo2083
375 http://commons.wikimedia.org/wiki/User:Alejo2083
376 http://commons.wikimedia.org/wiki/User:Alejo2083
377 http://commons.wikimedia.org/wiki/User:Alejo2083
378 http://commons.wikimedia.org/wiki/User:Alejo2083
379 http://commons.wikimedia.org/wiki/User:Alejo2083
380 http://commons.wikimedia.org/wiki/User:Alejo2083
381 http://commons.wikimedia.org/wiki/User:Alejo2083
382 http://commons.wikimedia.org/wiki/User:Alejo2083
383 http://commons.wikimedia.org/wiki/User:Ambrevar
384 http://commons.wikimedia.org/wiki/User:Ambrevar
385 http://commons.wikimedia.org/wiki/User:Ambrevar
386 http://en.wikibooks.org/wiki/en:User:Arnehe
387 http://en.wikibooks.org
388 http://en.wikibooks.org/wiki/en:User:Arnehe
389 http://en.wikibooks.org
390 http://en.wikibooks.org/wiki/en:User:Arnehe
391 http://en.wikibooks.org
392 http://en.wikibooks.org/wiki/en:User:Arnehe
393 http://en.wikibooks.org
394 http://en.wikibooks.org/wiki/en:User:Arnehe
395 http://en.wikibooks.org
396 http://commons.wikimedia.org/wiki/User:Alejo2083

206	Alessio Damato	GFDL

67. Licenses

67.1. GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion. 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work. 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary. 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures. 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee. 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date. * b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices". * c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it. * d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate. 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange. * b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge. * c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b. * d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements. * e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying. 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or * b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or * c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or * d) Limiting the use for publicity purposes of names of licensors or authors of the material; or * e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or * f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way. 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10. 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so. 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it. 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law. 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program. 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such. 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version. 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

67.2. GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies. 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

* A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. * B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. * C. State on the Title page the name of the publisher of the Modified Version, as the publisher. * D. Preserve all the copyright notices of the Document. * E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. * G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. * H. Include an unaltered copy of this License. * I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. * J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. * K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. * L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. * M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. * N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. * O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements". 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate. 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document. 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with … Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

67.3. GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

* a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or * b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

* a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

* a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the Combined Work with a copy of the GNU GPL and this license document. * c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. * d) Do one of the following: o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

* a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.