

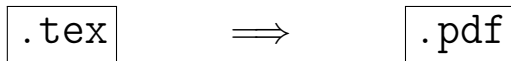
Creating Documents with T_EX/L^AT_EX

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n$$

L^AT_EX has a large German community

- The L^AT_EX Companion; Frank Mittelbach, Michel Goossens
- L^AT_EX Der typographische Einstieg; Tobias Berndt
- DANTE e.V.: <http://www.dante.de> and Books there, especially by Herbert Voß, e.g.
 - Typesetting mathematics with L^AT_EX; Herbert Voß
 - Typsetting tables with L^AT_EX; Herbert Voß
- L^AT_EX GE-PACKT; Karsten Günther
- L^AT_EX kurz & gut; Matthias Kalle Dalheimer

T_EX is a **typesetting system**, which similar to a compiler reads in a **source file** (with file extension `.tex`) and creates from that a formatted document file (in general a `.pdf` file \rightarrow pdf_latex).



```
\documentclass{scrartcl}
```

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

```
\begin{document}
```

```
\section{Introduction}
```

```
\TeX{} is a typesetting system  
by Donald Knuth.
```

```
\end{document}
```

1 Introduction

T_EX is a typesetting system by Donald Knuth.

Definitions

T_EX

the original typesetting system developed
by Donald Knuth, page description language (cf. HTML)

L^AT_EX

macro extensions for T_EX by Leslie Lamport,
document classes, packages

pdf_latex

program for L^AT_EX for direct output of pdf,
supports enhanced microtypography and
jpeg/pdf graphic files

Applications of L^AT_EX:

- publications: papers, books(!), master's theses etc.
- exercise sheets, forms, ...
- presentations: this one (documentclass beamer), slides
- letters (also serial letters)
- vector graphics with help of PSTricks
- posters

\LaTeX example

```
\documentclass{scrartcl}

\begin{document}

Hello, world! % document text

\end{document}
```

Compilation with help to the **shell command** (Caution: no GUI program!)

```
pdflatex helloworld.tex
```

Result is a file **helloworld.pdf**.

`\` – backslash starts a T_EX command

`{ }` – curly braces enclose mandatory arguments of T_EX commands. In a text, `{ }` can be created via `\{` and `\}`.

`\documentclass` – defines the document style, **always as 1. line!**, basic settings for paper format, font size, font, etc. (only in L^AT_EX), begin of the preamble

`scrartcl` – KOMA script class for short documents in DIN A4

preamble – the section between `\documentclass{}` and `\begin{document}`, settings for the whole document

`\begin{document}` – end of the preamble, begin of the actual document

`\end{document}` – end of the document, everything after will be ignored

`%` – percent sign starts a comment, if % should be used in the text: `\%` (backslash directly before percent sign)

If `pdflatex` encounters an unknown \LaTeX command, e.g., misspelled, it pauses the compilation process. E.g., if `\nd{document}` was written instead of `\end{document}`:

```
\! Undefined control sequence.  
1.6 \nd  
      {document}  
?
```

In this case `x`, Return will cease the whole compilation process.

Tip: `h`

Instead of `x` one can first try to get some help with `h`. Unfortunately, this is only sometimes helpful.

Suggestions for editing .tex files with an editor program

- Almost every editor recognizes T_EX files by their file extension .tex (e.g., hallowelt.tex) and highlights then T_EX commands optically (e.g., by a different font/color).
- Therefore, the name of the file should **always** be passed as an argument when starting the editor:

```
emacs helloworld.tex &
```

- The file name should **not contain any blanks!** (Why?)

Aufgabe 5.1 A simple \LaTeX document

- 1 Create a new directory for your \LaTeX documents.
- 2 Change to this directory and create with help of an editor, e.g., emacs, a file `helloworld.tex` with the content shown in the example above. Compile the file with `pdflatex`. View the output PDF file with `okular`:
`okular helloworld.pdf &`

Info: Viewing PDF files

If you want to create a PDF document by yourself, i.e. it will possibly change while viewing, it is recommended to use `okular` or `evince` instead of `acroread`. The PDF viewers `okular` and `evince` reload the PDF file automatically and update the view, if the file has changed.

Including packages I

The functionality of \LaTeX is fundamentally enhanced by the use of *packages*. The \LaTeX command

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

includes the package `babel` with the option `english` (also possible: `ngerman`): language dependent settings for documents, e.g., *Inhaltsverzeichnis* (`ngerman`) instead of *table of contents* etc.

Packages must be included in the **preamble**, i.e. **after** `\documentclass{ }` and before `\begin{document}`:

```
\documentclass{scrartcl}  
  \usepackage[ngerman]{babel}  
\begin{document}
```

Squared brackets []

Optional arguments of \LaTeX commands are put in squared brackets `[]`.

The actual text of the document

- is entered as running text; line breaks, multiple space characters and blank lines are usually ignored by \LaTeX
- a **blank line** enforces a new **paragraph**, i.e. including an *indent* of the next line
 - other formatting is done automatically by \LaTeX
or by \LaTeX commands

Hint: Input of umlauts and other non-ASCII characters

Current versions of \LaTeX can handle UTF-8 and therefore can also process umlauts etc. For older \LaTeX compilers (before 2018) the package `inputenc` will be needed otherwise:

```
\usepackage[utf8]{inputenc}
```

- a double backslash `\\` results in a line break (if allowed by \LaTeX at this position)

Problems with blank lines

Under certain circumstances, \LaTeX does not allow a line break via `\\`, e.b., directly after *environments*, if these already imply a line break. In this case an error message is issued during compilation:

```
! LaTeX Error: There's no line here to end.
```

If there should be still inserted a vertical space, then this can be achieved by, e.g.:

```
\vspace{0.2cm}
```

- by default \LaTeX uses (full) justification for the text output. Therefore, \LaTeX performs automatically hyphenation with the help of elaborated algorithms.

Aufgabe 5.2 Fully justified text

Convince yourself of the above mentioned properties by including both packages `babel` and `inputenc` in your document.

Add also a sufficiently large text to your document to see the hyphenation in the output PDF, e.g., via *copy & paste*.

Hints:

- To visualize the hyphenation better, one can shrink the *type area*:
`\documentclass[DIV=4]{scrartcl}`
Then, also use the option `\sloppy` in the preamble.
- `nedit` and some other editors do **not** support UTF-8 encoding, better use, e.g., `emacs`.

Excursus: Character encoding

- Unicode: int. standard, for assigning a digital code to *every* character, 7×2^{16} code points, e.g., U+00DF for ß
- ASCII (American Standard Code for Information Interchange): 7 bit (=128 characters) encode English alphabet and some special characters (starting with space) → universal minimum standard
- ISO-8858-1 (Latin-1): 8 bit (=256 characters) allows for additional characters on top of ASCII (= first 128 characters), e.g., umlauts → family of encodings
- UTF-8 (Unicode Transformation Format): encoding for unicode, based on 1 up to 4 byte (each 8 Bit), first 128 characters correspond to ASCII → de-facto standard of the internet
- UTF-32: encoding for all unicode characters with a fixed length of 4 byte → minimum programming effort vs. waste of space (4× larger documents than with Latin-1)

Tip: For compatibility reasons a \LaTeX document should only contain ASCII characters. One can obtain all special characters in the output by corresponding \LaTeX commands:

<code>\"a</code>	\rightarrow	ä	<code>\'{\i}</code>	\rightarrow	í	<code>\~{n}</code>	\rightarrow	ñ
<code>\"o</code>	\rightarrow	ö	<code>\' {e}</code>	\rightarrow	é	<code>\c{c}</code>	\rightarrow	ç
<code>\"u</code>	\rightarrow	ü	<code>\`{a}</code>	\rightarrow	à	<code>\v{c}</code>	\rightarrow	č
<code>\ss{}</code>	\rightarrow	ß	<code>\^{o}</code>	\rightarrow	ô	<code>\AA</code>	\rightarrow	Å

and many more

Quotation marks

\LaTeX can print out all kinds of quotations marks, but they must be given explicitly, e.g.,

typographic	<code>` `</code> (2 slanted)	<code>“H</code>
	<code>"</code> (2 straight)	<code>H”</code>
German (96)	<code>"`</code> (double + slanted)	<code>“‘H</code>
	<code>"'</code> (double + straight)	<code>H”’</code>

\LaTeX special characters

Some characters have a special meaning in the \LaTeX source code and must be “escaped” with a backslash `\`

<code>\textbackslash</code>	\rightarrow	<code>\</code>	<code>\&</code>	\rightarrow	<code>&</code>
<code>\#</code>	\rightarrow	<code>\#</code>	<code>\\$</code>	\rightarrow	<code>\$</code>
<code>\</code>	\rightarrow	<code>(blank)</code>	<code>\%</code>	\rightarrow	<code>%</code>

Structure of a document I

As a typesetting system, \LaTeX allows automatic control of special text elements like headlines, captions, formulae, tables, citations etc.

The essential structure of a document is done by commands like

In the .tex file

```
\section{Introduction}
```

The command `\section{ }` causes:

- highlighting/formatting as a headline
- setting a counter for the corresponding indenture level
- creating an entry in the table of contents

Output in the .pdf file

1 Introduction

There exist the following predefined levels of sections (descending):

- 1 part
- 2 chapter (not in `scrartcl`)
- 3 section
- 4 subsection
- 5 subsubsection
- 6 paragraph
- 7 subparagraph

The table of contents

For a document with indenture levels, the command

```
\tableofcontents
```

can create automatically a table of contents at the corresponding position in the document, in general in the beginning, but **after** `\begin{document}`. In this table of contents only the upper levels of sections are shown by default.

If a new page should be started afterwards, one can use

```
\newpage
```

This command also works at other locations in the document.

Aufgabe 5.3 Document structure and table of contents

Subdivide your \LaTeX document (sufficiently long text) with levels of sections, use meaningful headlines.

Create a [table of contents](#). You need to run `pdflatex` at least **twice** to get a correct table of contents (Why?).

If you define in the preamble by

```
\author{Hellen Keller}
```

an author and by

```
\title{The Frost King}
```

a title, you can use

```
\maketitle
```

```
\newpage
```

in the document (after `\begin{document}`), to make a [front page](#). The date can be manipulated by the command

```
\date{1891}
```

During the compilation of the `.tex` file additional files, besides the output PDF file, are created:

- `.toc`
table of contents
- `.aux`
commands for the structure of the document,
(literature) citations, etc.
- `.log`
information by the \LaTeX compiler (versions, packages, errors)

\LaTeX updates these files in every compiler run and uses informations from the documents to do so. Therefore, one needs to compile twice for a consistent table of contents.

Unordered and ordered lists are realized in \LaTeX with help of [environments](#):

```
\begin{environment}
```

```
\end{environment}
```

A simple unordered list (bullet points, items) can be created by using the environment `itemize`:

```
\begin{itemize}
```

```
  \item a thought
```

```
  \item another thought
```

```
\end{itemize}
```

Result:

- a thought
- another thought

To create an ordered list or enumeration, it is recommended to use the environment `enumerate`:

```
\begin{enumerate}  
  \item first item  
  \item second item  
\end{enumerate}
```

Result:

- ➊ first item
- ➋ second item

It is possible to nest up to four levels, for each numbering a different type of count ist used:

- Arabic numerals: 1.
- lowercase letters: a)
- Roman numerals (lowercase): i.
- capitals: A.

Aufgabe 5.4 Lists

Create a \LaTeX document with an **unordered list** (e.g., shopping list) and an **ordered list** (e.g., contents of this lecture). Try to make nested lists:

```
\begin{enumerate}
  \item first item
    \begin{enumerate}
      \item first subitem
    \end{enumerate}
\end{enumerate}
```

center and quote are also important environments

Center text

```
\begin{center}  
When I had journeyed half of our life's way,  
  
I found\\  
myself within a shadowed forest,\\  
for I had lost the path that does not stray.  
\end{center}
```

Result:

When I had journeyed half of our life's way,
I found
myself within a shadowed forest,
for I had lost the path that does not stray.

(Dante, Divine Comedy)

The environment `verbatim` is used for literal text, including \LaTeX special characters like `\`, `{}`; e.g., for source code:

```
#include <iostream>
using namespace ::std ;
int main (){ return 0 ; }
```

is created by

```
\begin{verbatim}
#include <iostream>
using namespace ::std ;
int main (){ return 0 ; }
\end{verbatim}
```

Quotations are separated from the justified text with the environment quote:

Quotations

```
\begin{quote}  
  All spirits are enslaved  
  which serve things evil:\\  
  Thou knowest if Jupiter be such or no.  
\end{quote}
```

Result:

*All spirits are enslaved which serve things evil:
Thou knowest if Jupiter be such or no.*

(Shelley, Prometheus Unbound)

Tables can be typeset with help of the environment `tabular`. This environment requires the specification of *mandatory arguments* in curly braces, namely the number and alignment of the table columns:

```
\begin{tabular}{lcr}  
city & country & river \\ % linebreak necessary  
Yaound\`e & Yemen & Yukon River  
\end{tabular}
```

Result:

city	country	river
Yaoundé	Yemen	Yukon River

Meaning of the column arguments:

<code>l</code>	left-aligned
<code>c</code>	centered
<code>r</code>	right-aligned

Often it is required to number consecutively all tables of a document and to place a caption *above* the table. That can be done in the table environment. `table`.

```
\begin{table}[h] % tries, to put the table [h]ere
\begin{center}      % recommendend
\caption{Taxonomy in biology}
\begin{tabular}{lcr}
  regnum & divisio & subphylum \\
  kingdom & division & subphylum/subdivision
\end{tabular}
\end{center}
\end{table}
```

Result:

Table: Taxonomy in biology

regnum	divisio	subphylum
kingdom	division	subphylum/subdivision

Options for positioning of tables

Tables are *floating elements* in \LaTeX , i.e., they cannot be broken over a page, and therefore do not necessarily appear at the same position in the output as in the input. Instead their position in the output document is determined by typographic rules, in general they are placed at the beginning of the page.

However, one can influence the automatic positioning of tables:

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

[h]	here, if possible
[t]	top (of the page)
[b]	bottom (of the page)
[p]	page, put alle tables on an extra page extra
[!]	overwrite float restrictions
[H]	Here! (unconditional) only with package <code>float</code>

The default is `[tbp]`, recommended is `[htbp]`. The parameter `H` (`\usepackage{float}`) enforces the positioning at the exact position of appearance on the cost of a scrambled layout.

Lines in tables

Sometimes tables are better readable by the use of lines.

vertical lines between columns

```
\begin{tabular}{l|ll|l}
```

→ straight line | (*pipe*) between the columns

horizontal lines between rows

```
\hline
```

→ after a linebreak

Excursus: typographic tables

beautiful tables have only horizontal lines of different thickness. That can be done by including the package

```
\usepackage{booktabs}
```

Instead of `\hline` one uses only

type of line	position
<code>\toprule</code>	top, before table head
<code>\midrule</code>	after table head, in the table body
<code>\bottomrule</code>	bottom, after the last line

Aufgabe 5.5 Typesetting tables

Use \LaTeX to typeset and complete the following table:

language	type	comments by	meaning of semicolon
shell	script		multiple commands per line
gnuplot			
\LaTeX			
C++			

References to tables

The numbering, which is assigned by \LaTeX automatically to the table environment, can get an **internal** name via `\label{name}` (directly after `\caption{ }`). This internal name can be referenced in the text with `\ref{name}`, e.g.:

```
...  
\caption{Computer languages}  
\label{tab:programminglanguages}  
...  
\end{table}
```

The examples listed in Tab.~`\ref{tab:programminglanguages}`

Aufgabe 5.6 References in \LaTeX

Create with help of `\label` an internal name for your table and make a reference to this name in your text with `\ref`

To get this correctly working, you must run `pdflatex` **twice**.

What is the advantage of these reference over using literal numbers in the text?

Further usage of references

The usage of `\label{name}` and `\ref{name}` is not restricted to the `table` environment only, but also works for all `environments` that get a number, so also for `figures`.

Moreover, references also work with numbered structure elements, e.g., `sections` and `equations` (see below).

Hint: Naming scheme

You can choose any name for the references, however to get not too confused, put the type of object you want to reference in the beginning of the name, e.g., `tab:`, `fig:`, `sec:`, `eq:`, ...

If the package `graphicx` is included in the preamble

```
\usepackage{graphicx}
```

it is possible to include pictures from external files. Under `pdflatex` the following data formats are supported:

`pdflatex`: supported graphic formats

pdf

png

jpeg

The graphic file is included with the command

```
\includegraphics[width=x.x,height=y.y,angle=nnn]{file.pdf}
```

The specification of the width, height, and an rotation angle is optional. Allowed units for width and height are, e.g., cm, mm, pt (≈ 0.34 mm), `\textwidth`, `\textheight`. The rotation angle is given in degree (without unit), e.g.:

```
\includegraphics[width=\textwidth,%  
                height=4.5cm,angle=270]{file.pdf}
```

Aufgabe 5.7 Include a pdf graphic

Use the package `graphicx` and the command `includegraphics` to include the graphic file `wr-pz.pdf` in your `.tex` file. Try to make the graphic as wide as possible.

As for tables, there is an environment, which is used to embed graphics. This environment, `figure`, supports numbering and captions:

```
\begin{figure}[htpb!]  
\begin{center}  
\includegraphics{file.pdf}  
\caption{Impactparameter coordinates}  
\label{fig:pz-coord}  
\end{center}  
\end{figure}
```

Aufgabe 5.8 The `figure`-Environment

- 1 Embed the included graphic in a `figure` environment.
- 2 Reference this figure in the text.

Hint

With help of the command

```
\listoffigures
```

a list of figures can be automatically created in the \LaTeX document (similar to list of tables).

Analogously, the command

```
\listoftables
```

creates a list of all tables in the document.

We already know: The program `pdflatex` allows direct PDF output from a `.tex` file. Originally, the program `latex` created only files in the DVI format (device independent). DVI files can be displayed with DVI viewer programmes, and can be converted to PostScript files.

Aufgabe 5.9 DVI output

Compile a `.tex` file without a figure (e.g. `myfile.tex`) via
`latex myfile.tex`

Try to open the output `.dvi` file.

Hint: `latex` can only handle graphics in the PostScript format (`eps`, `ps`).
DVI files do not contain graphics, but only links to them, and are therefore relatively small.

In a next step, the `.dvi` file can be converted to a PostScript file with the program `dvips` (`dvi`→`ps`):

Aufgabe 5.10 The program `dvips`

Run the program `dvips`, it needs the name of the `dvi` file (from the last task) as argument, e.g.,
`dvips myfile.dvi`
Open the output PS file.

The resulting PS file contains now all essential elements, e.g., graphics, and can be printed directly via `lpr` .

Aufgabe 5.11 PDF output

With help of the program [ps2pdf](#) PostScript files can be converted to PDF files

`ps2pdf myfile.ps`

Compare the resulting PDF with the one directly created with `pdflatex`.

latex	vs.	pdflatex
dvi	output	pdf
yes!	PS functions	no
eps, ps	supported graphics	jpeg, pdf, png
limited	microtyping	package microtype

Besides the already mentioned document class `scrartcl` there are many more document classes. Especially from the KOMA script collection (for DIN formats):

- `scrartcl` → article
- `scrbook` → book (with `\chapter`)
- `scrreprt` → report
- `scrlettr` → letter

Also given are the corresponding standard classes for US paper format “letter”.

For the class `scrbook`, twoside layout is the default, while for the other classes it is oneside. That can be changed:

```
\documentclass[twoside]{scrartcl}
```

The most important option of the KOMA script classes is the DIV value for the calculation of the type area (German “Satzspiegel”):

```
\documentclass[DIV=15]{scrartcl}  
% very small margins; always 3 stripes for the margins
```

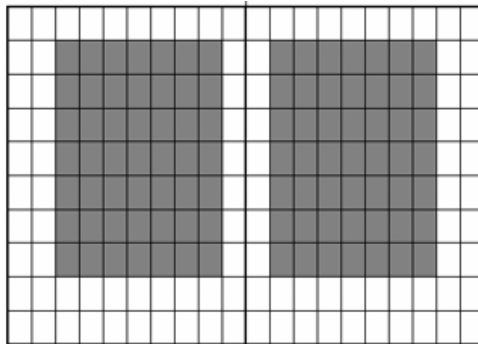


Figure: Construction of the type area via classical “Neunerteilung” (dividing by nine)

Excursus: Construction of the type area with the KOMA-Script classes

The option `[DIV=calc]` enables automatic calculation for other formats than DIN A4, e.g., when using the option `der Option [a5paper]`. A construction of the type area by a medieval method is possible by the option `[DIV=classic]`. *After* inclusion of some packages (e.g., for alternative fonts) a recalculation of the type area might be necessary via

```
\typearea[current]{calc}
```

The so-called binding correction is specified with the option `[BCOR=length]`.

Example:

```
\documentclass[a5paper,DIV=classic,BCOR=5mm]{scrreprt}  
\usepackage{bookman}  
\typearea[current]{calc}
```


More document classes:

- `beamer`: screen presentations (this one)
- `aa`: publication style for Astronomy & Astrophysics
- `sciposter`: posters, e.g., in A0 format

and many more

The standard settings of \LaTeX and the document classes, like `scrartcl` are usually optimal and should not be modified arbitrarily. However, it is possible to make modifications.

Font size

can be defined for the whole document in unit points:

```
\documentclass[11pt,DIV12]{scrbook}
```

Moreover, the font size can be always changed relative to this value:

```
{\Large{}}A headline\\
```

The modification is only within the curly braces.

A headline

The modification is only within the curly braces.

There are the following commands to change the font size (in general each for at least 1pt difference in size)

<code>\tiny</code>	aA
<code>\scriptsize</code>	aA
<code>\footnotesize</code>	aA
<code>\small</code>	aA
<code>\normalsize</code>	aA
<code>\large</code>	aA
<code>\Large</code>	aA
<code>\LARGE</code>	aA
<code>\Huge</code>	aA

Changing the font type and appearance

Depending on the document class, there is a default font type, e.g., Computer Modern with serifs for `scrartcl`.

The appearance can be changed as follows:

<code>\sffamily</code>	Sans Serif – without serifs
<code>\rmfamily</code>	Roman – normal font with serifs
<code>\ttfamily</code>	TypeWriter – monospace font with broad serifs
<code>\slshape</code>	<i>slanted</i> – slanted font
<code>\scshape</code>	SMALL CAPITALS – small caps
<code>\bfseries</code>	Bold Face – bold text
<code>\emph{}</code>	<i>emphasize</i> – usually italic, slanted font

Certain combinations are possible:

```
{\bfseries\sffamily{}}Bold without serifs}
```

Bold without serifs

One of the biggest strengths of \LaTeX is the typesetting of formulae. Formulae can be typeset in two ways:

- in a text line, e.g., $\sum \frac{1}{x^2}$ via:

... , e.g., `\sum \frac{1}{x^2}` via:

- or separately with a consecutive numbering:

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad (1)$$

The formula for the Riemann zeta function of 2 was created so:

```
\begin{align} % !!! requires \usepackage{amsmath}
\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2}
         = \frac{\pi^2}{6}
\end{align}
```

Aufgabe 5.12 Typesetting formulae

Write in your \LaTeX document the formula for calculating Euler's number as an infinite series in a text line and as separated formula (package `amsmath` required).

Hint: The packages `amsmath`, `amsfonts`, `amssymb` extend the capability for representation of (mathematical) symbols enormously, e.g., `\checkmark` = ✓, `\lessssim` = \lesssim

Moreover, the package `amsmath` enables the use of the command `\text{ }` within math environments to typeset **normal text**, e.g., **for physical units**:

$$U = 230 \text{ V}$$

through: `U = 230\,\text{V}`

Also use `\text{ }` for certain subscripts that are not indices, e.g., M_{ini} , v_{rad}

The `align` environment also allows formulae with multiple lines with a tabular-like internal formatting using `&`:

```
\begin{align}
\zeta(2) &= \sum_{n=1}^{\infty} \frac{1}{n^2} \\
&= \frac{\pi^2}{6} \label{eq:pi}
\end{align}
```

Caution: No blank lines in `align`!

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} \tag{2}$$

$$= \frac{\pi^2}{6} \tag{3}$$

The lines have consecutive numbers, which can be referenced with `label/ref`, the usual format is: Eq. (3).

The numbering can be switched off by using `*` in the name of the environment:

```
\begin{align*}
\sin ^2 + \cos ^2 = 1
\end{align*}
```

$$\sin^2 + \cos^2 = 1$$

Hint: Exactly as for units (V, kg, cm, etc.) it is common to typeset names of functions (sin, cos, tan, etc.) in formulae as **straight** normal text. This can be done with the help of predefined macros, e.g., `\sin`, `\cos`, etc.

Aufgabe 5.13 Multiline Formulae

Typeset a multiline formula and reference it in your text (Remember that you have to run `pdflatex` twice.)

Excursus: Formula environments

Actually the dollar signs `$... $` used for [inline formulae](#) are an abbreviation for `\begin{math} ... \end{math}`. An alternative abbreviation is `\(... \)`.

[Single-line offset](#) formulae can be typeset in \LaTeX via

`\begin{displaymath} ... \end{displaymath}` or with their corresponding abbreviation `\[... \]`, but [without](#) a consecutive [numbering](#). Moreover, there is also an offset formula environment that already exists in \TeX and is enclosed by double dollar signs: `$$... $$` (not recommended).

[Single-line offset](#) formulae [with consecutive numbering](#) are also possible via

`\begin{equation} ... \end{equation}`.

Apart from the `align` environment from the `amsmath` package, there exists also the `eqnarray` environment for multiline offset formulae, however its horizontal spacing is wrong and it is therefore not recommended:

$x = 3$	<code>equation</code>	(4)
$x = 3$	<code>align</code>	(5)
$x = 3$	<code>eqnarray</code>	(6)

(source code for `eqnarray`: `x & = & 3`, hence internally three columns, instead two as for `align`.)

There are many more possibilities to typeset formulae, e.g,

```
e = \lim_{n\rightarrow \infty}
\left( 1 + \frac{1}{n} \right)^n
\nonumber
```

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n$$

The command `\nonumber` switches off the numbering for the current line in the formula.

The commands `\left` and `\right` followed by a corresponding brace `()` `[]` `\{ \}` adjust the height of the braces to the content.

Another example:

```
H(x) = \left\{
    \begin{array}{ll}
      0: & x < 0 \\
      1: & x \geq 0
    \end{array}
\right.
```

will represent the piecewise Heaviside function:

$$H(x) = \begin{cases} 0 & : x < 0 \\ 1 & : x \geq 0 \end{cases} \quad (7)$$

In this example `\right.` works as an invisible mark for the adjustment of the braces. The environment `array` is similar to the tabular environment and can also be used to display $n \times m$ matrices.

Some more symbols and formattings:

$$\begin{array}{ll}
 \dot{L} \stackrel{!}{=} 0 & \texttt{\dot{L} \stackrel{!}{=} 0} \\
 \sqrt{\sqrt{16}} = \sqrt[4]{16} & \texttt{\sqrt{\sqrt{16}}} = \texttt{\sqrt[4]{16}} \\
 \int_M d\omega = \oint_{\partial M} \omega & \texttt{\int_M d\omega = \oint_{\partial M} \omega} \\
 e = \underbrace{\frac{1}{0!}}_{0!=1} + \dots & \texttt{e=\underbrace{\frac{1}{0!}}_{0!=1}+\ldots}
 \end{array}$$

- `hyperref`

```
\usepackage{hyperref}
```

→ creates clickable links in PDF documents, e.g., table of contents, references, URLs

```
\url{https://www.astro.physik.uni-potsdam.de}
```

<https://www.astro.physik.uni-potsdam.de>

(You can click on the link above!)

Caution!

If used, the package `hyperref` must be included as the **last** package in the preamble.

- PStricks

```
\usepackage{pstricks}
```

→ PostScript graphic via \LaTeX commands (**requires latex + dvips**)

```
\begin{pspicture}(-1,-1)(3,1)  
  \psline{<->,linecolor=blue}(0,0)(2,0.5)  
\end{pspicture}
```



Other examples, packages, and manuals at:

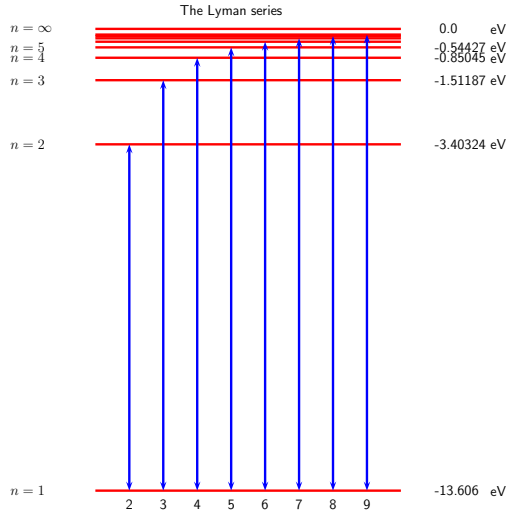
```
http://tug.org/PSTricks/main.cgi/
```

- `multido`

```
\usepackage{multido}
```

→ simple loops for reptition of commands

```
\multido{\i= 2 + 1 }{8}  
{  
  \setlength{\xdist}{1.0cm * \real{\i}}  
}
```



Example: `pstricks` + `multido` (`lymanseries.tex`)

For \LaTeX , there exist a convenient system to create bibliographies, called `bibtex`.

To use this, write reference into a file with extension `.bib`

Format of bib entries

```
@ARTICLE{cak1975,  
  author = {{Castor}, J.~I. and {Abbott}, D.~C.  
           and {Klein}, R.~I.},  
  title = "{Radiation-driven winds in Of stars.}",  
  journal = {ApJ},  
  year = 1975,  
  month = jan,  
  volume = {195},  
  pages = {157-174},  
}
```

In the first line, the *internal name* for referencing is given (here: `cak1975`). This name can then be referenced in the document text: `\cite{cak1975}`.

The bibliography within the document is created so:

```
\bibliography{mybibfile} % without extension .bib
```

typically at the end of the document.

To get the references working, the [program bibtex](#) must be called, with the name of the .aux file of the corresponding T_EX document as mandatory argument:

```
bibtex texdocument.aux
```

Subsequently, pdf_latex must be executed **twice**.

Hint: For convenience, it is recommended to use bibtex within a L^AT_EX development environment, like TeXnic-Center, TeX-Shop or to use shell scripts.

Within the text referencing is done with the command

Example for citation

```
\cite{cak1975}
```

At the corresponding position in the document the reference is shown and added to the bibliography. How the reference is displayed must be defined in the preamble via

Bibliography style definition

```
\bibliographystyle{plain}
```

There are many different styles available. Journals (e.g., A & A) often define their own style.

Example

An analytic solution for line driven winds is presented in [1].

Bibliography

- [1] J. I. Castor, D. C. Abbott, and R. I. Klein.
Radiation-driven winds in Of stars.
ApJ, 195:157–174, January 1975.

Aufgabe 5.14 Using BibTeX

Create a .bib file for your document, insert the example entry and cite it in your document. Remember to include also the necessary commands `\bibliographystyle{ }` und `\bibliography{ }` as well as the execution of `bibtex` and `pdflatex`.

Summary:

The .bib file

```
@ARTICLE{NameYear, ... }
```

In the .tex document

```
...  
\bibliographystyle{plain} % or: alpha, apalike, ...  
\begin{document}  
...  
\cite{NameYear} % label as in .bib-Datei  
...  
\bibliography{...} % name of the .bib file w/o extension .bib
```

- 1 pdflatex document.tex
- 2 bibtex document.aux
- 3 pdflatex document.tex
- 4 pdflatex document.tex

While the plain bibliography style is quite compact and widely used in physics, in astrophysics a different, more readable style is common:

Excursus: Bibliography style in astrophysics

In the preamble:

```
\usepackage{natbib}  
\bibliographystyle{mnras}
```

The natbib package brings `\citet{}` and `\citep[] []{}`, the resulting reference in the text looks for `\citet{cak1975}` and for `\citep[e.g.,][and references therein]{cak1975}` as follows:

... as [Castor et al. \(1975\)](#) helps to ...

Many authors ([e.g., Castor et al., 1975, and references therein](#)) have ...

And the bibliography entry (entries will be listed in alphabetical order):

Castor J. I., Abbott D. C., Klein R. I., 1975, ApJ, 195, 157

There are many more possibilities to structure the document

- footnotes¹ by `\footnote{ }`
- appendix via `\appendix`
- index using the program `makeindex file` and the commands: `\makeindex` (in preamble), `\index{ }` (in text) and finally `\printindex`

Spell check using `ispell`

```
ispell -t -d english helloworld.tex
```

Microtypography

- the microtypographic capabilities of pdf \TeX are enabled by just using the package `microtype`

```
\usepackage{microtype}
```

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen sie in Buchstabhausen an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort und versorgt sie mit den nötigen Regelialien. Es ist ein paradiesmatisches Land, in dem einem gebratene Satzteile in den Mund fliegen. Nicht einmal von der allmächtigen Interpunktion

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen sie in Buchstabhausen an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort und versorgt sie mit den nötigen Regelialien. Es ist ein paradiesmatisches Land, in dem einem gebratene Satzteile in den Mund fliegen. Nicht einmal von der allmächtigen Interpunktion werden die Blindtexte beherrscht.

left: without package microtype

right: with microtype, especially with optical margin alignment for *apparent* smooth text margin (right)

- sometimes it is necessary to do manually microtypographic corrections:

command	result	example
<code>\kernx</code>	inserts a space x (also negative!)	<code>\kern1mm</code>
<code>\,</code>	inserts a half space	<code>\$U=230\$\,,V</code>
<code>\!</code>	inserts a small negative space	<code>V\!ong</code>

Vong *geometrically* equidistant without kerning

Vong *optically* equidistant with kerning

¹Footnotes are useful for, e.g., tables.

Disadvantages of using \LaTeX :

- default output is the unusual DVI (device independent file format):
alternative `pdflatex` makes directly PDF files
- high threshold for beginners caused by the abstraction of the typesetting (no WYSIWYG)
- some automatism of the typesetting limit the capabilities for free scope of design (no *real* layouting/DTP)
- some packages are only supported by `latex`, others only by `pdflatex`, e.g., `pstricks` and `microtype`; inconsistent settings by different packages
- although PDF output now common, not all PDF capabilities are supported, e.g., media embedding depends on operating system

Strengths of \LaTeX :

- free and available on all systems $\rightarrow \text{\TeX Live}$
- separation of form and content (unfortunately not complete)
- therefore high reusability & flexibility
- perfectly typeset formulae and documents (e.g. letters)

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

- automatism for table of contents, bibliography, hyperlinks, ...
- no length restrictions for documents, changes can be done at any position, stable, low memory requirements
- popular – in many areas the de facto default