
Typesetting with T_EX / L_AT_EX

An Introduction

F. C. Langbein

School of Computer Science
and Informatics
Cardiff University





What is T_EX / L^AT_EX?

- T_EX is a computer *typesetting system*
(created by D. Knuth while writing “The Art of Computer Programming”)
 - T_EX is really a complete *programming language* aimed at creating documents
 - Basic T_EX language can be expanded by macros
 - *Formats* are large macro sets for layout, formatting, etc.
- L^AT_EX is a *particular T_EX format* created by Leslie Lamport
 - Set of macros to specify documents on a high level
 - In many ways L^AT_EX is similar to SGML/XML/HTML
 - Really *L^AT_EX 2_ε*, an intermediate step towards L^AT_EX3



Why L^AT_EX?

- You can easily create beautiful, long, complex documents
 - It *knows* a lot about typesetting
 - It produces *exact* rather than approximate results
(no ugly WYSIWYG system, but WYMIWYG)
 - Easy to *modify and expand* using macros
 - A set of *well-designed tools* to prepare documents
(no big expensive software package that often fails in mysterious ways)
- Supported on (nearly) all computer platforms
 - Free and commercial versions available
 - T_EX document sources are in *plain text* (ASCII)



Why L^AT_EX?

- Suitable for large projects
 - Clear, explicit *separation between content and style*
 - *Ease of generating tables* of contents, figures, bibliography, references, indices, . . .
 - Ability to *split long documents* (books, etc.) into smaller, more manageable pieces
 - Can easily be combined once writing is complete
- *Stability* and *interchangeability* of documents (de-facto standard for scientific documents)
- No upgrade pressures
 - *Converges* to stable, feature-complete version (instead of adding ever more bugs and useless features)

TEX / L^AT_EX Resources

➤ Web sites

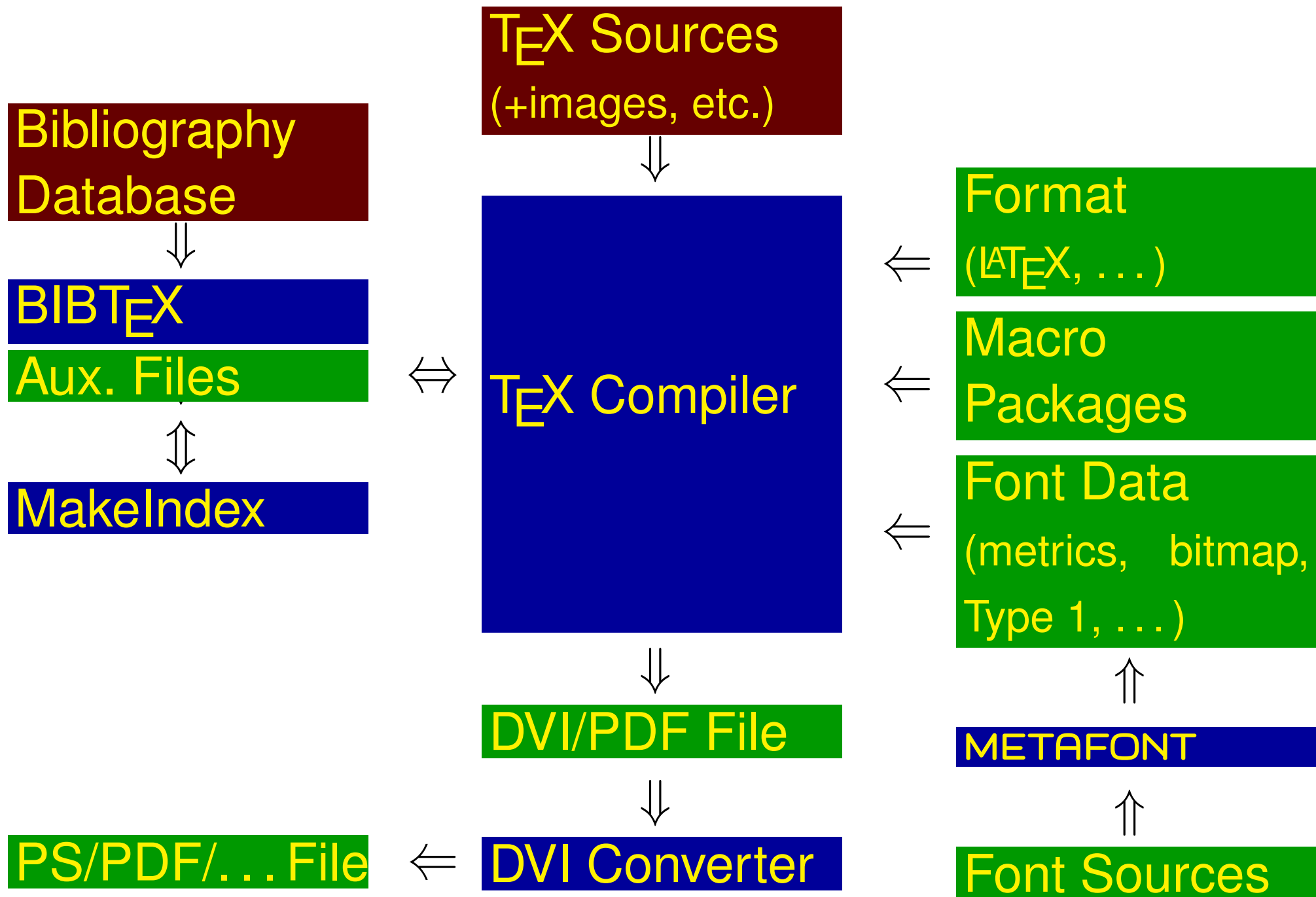
- <http://www.langbein.org/teaching/typesetting>
- Comprehensive T_EX Archive Network (CTAN) at <http://www.ctan.org/>
- T_EX Users Group (TUG) at <http://www.tug.org/>

➤ T_EX newsgroup at `comp.text.tex`

➤ The L^AT_EX Companions

- H. Kopka, P. Daly. A Guide To LaTeX, 4th ed, 2003.
- F. Mittelbach, M. Goossens. The L^AT_EX Companion, 2nd ed, 1994.
- M. Goossens, F. Mittelbach, et al. The L^AT_EX Graphics Companion, 2nd ed, 2007.

Components of the T_EX System



Overall Document Structure

➤ Every \LaTeX document has the following structure

```
\documentclass [options] {class}  
preamble  
\begin{document}  
    document body  
\end{document}
```

- **class**: *document type*, e.g. article, report, book
- **options**: optional list of document type *modifications*
- **preamble**: contains *formatting, layout, etc.* information and inclusion of additional macro packages
- **document body**: actual *contents* of the document



Document Example

```
\documentclass[a4paper,12pt]{article}

\usepackage{times}

\title{A Sample Document}

\author{J.\ Doe \and A.\ Foo}

\date{\today}

\begin{document}

\maketitle

\section{Introduction}

Lorem ipsum dolor sit amet, consectetur
adipisicing elit...

\end{document}
```

L^AT_EX Commands

➤ Three versions of commands

- *Non-letter characters* &, \$, %, ~, _, {, }, #, ^ have special meaning telling L^AT_EX to do something
- *Backslash* \ followed by *single non-letter character*
(e.g. any of the above characters can be produced by adding \)
- *Backslash* \ followed by *one or more letters*
- Letter/non-letter characters are defined explicitly!
(i.e. they may be redefined sometimes)

➤ Commands generally have the structure

```
\COMMAND [optarg] {mandarg}
```

- There may be none, one or multiple optional / mandatory arguments

Environments

- An *environment groups segments* of code
 - The body of an environment is treated differently from the “outside”
 - General syntax

```
\begin{environment}  
    body of environment  
\end{environment}
```

- For example, a `center` environment centring its body

```
\begin{center}  
    centred \scshape text  
\end{center}
```

```
text before environment  
    centred TEXT  
text after environment
```

- Changes inside an environment are usually local

Grouping

- Text can be *grouped* using { `some text` }
- Changes inside group are *local*
- For example, to emphasise the text in the group:

```
This is {\em important}.
```

This is *important*.

- Effect of command `\em` ends with end of group
- An argument to a command is similar to a group, but the command is outside, e.g.

```
This is \emph{important}.
```
- Yields the same result, but the *different* command `\emph` takes the text as argument

Characters, Words, Paragraphs

- \LaTeX regards *groups of characters separated by spaces* (even multiple spaces) or newlines as *words*
- A *blank line* (or multiple blank lines together) tells \LaTeX to begin a *new paragraph*
- `%` indicates a *comment* and everything until the end of line is ignored (incl. newline character)

```
This is an example  
paragraph.
```

```
The next parag% Comment  
raph starts here.
```

This is an example paragraph.

The next paragraph starts here.

- Other mechanisms for sentences, spacing, line and page breaks, special characters, ...



A Simple Example: `sample.tex`

```
\documentclass{article}
\begin{document}
% From Winged Death, H.P. Lovecraft and H. Heald
Here, in essence, is the text which the doctor read
aloud in that sinister [\dots] room [\dots]

\begin{center}
JOURNAL OF\THOMAS SLAUENWITE, M.D.
\end{center}

Touching {\em punishment} of Henry Sargent Moore,
Ph.D.\ of Brooklyn, New York, Professor of
Invertebrate Biology in Columbia University, New York,
N.Y. Prepared to be read after my death---for the
satisfaction of making public the accomplishment of
my revenge \dots
\end{document}
```



Compiling Documents

➤ To compile sources `sample.tex` under UN*X

```
> latex sample          # suffix .tex may be omitted
This is TeX Version 3.1415926
...
> xdvi sample          # view .dvi file under X11
> dvips sample         # generate standard ps file
> dvips -Ppdf -t a4 sample
                        # generate ps file for PDF / A4
> ps2pdf               # convert to pdf using
                        # ghostscript (Ver. 7 or later)

> pdflatex sample     # directly generate pdf file
                        # using pdflatex
```



Result of sample.tex

Here, in essence, is the text which the doctor read aloud in that sinister [...] room [...]

JOURNAL OF THOMAS SLAUENWITE, M.D.

Touching *punishment* of Henry Sargent Moore, Ph.D. of Brooklyn, New York, Professor of Invertebrate Biology in Columbia University, New York, N.Y. Prepared to be read after my death—for the satisfaction of making public the accomplishment of my revenge ...

Document Classes

- Document class determines the *overall structure* of the document
- Sets overall layout, formatting and basic behaviour of commands
- **Standard classes:** `article`, `report`, `book`, `letter`, `slides`, `(foils)`
- Classes usually provide similar commands, but typesetting results are different
- Special classes (for journals, special applications, etc.) exist
- Specify document class by:

```
\documentclass{article}
```

Common Class Options

- The class is *modified by options*
- Some common options
 - **Font size:** 10pt (default), 11pt, 12pt
sets basic font size and adjusts all other sizes
 - **Paper size / orientation:** a4paper, letterpaper, ..., portrait (default), landscape, oneside (default), twoside
 - **Columns:** onecolumn (default), twocolumn
- E.g. begin a document with

```
\documentclass[a4paper,12pt]{article}
```
- Options are class specific (see literature for details)

Packages

- New *special features* are added by *macro packages*
- Thousands of packages for all purposes available
- Include package in preamble by

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

- Package options modify default package behaviour
- Class options are forwarded to packages
- Inclusion order can be important, though it usually is not
- Available packages and classes depend on L^AT_EX installation

Sectioning commands

➤ Commands to *structure the document* defined by document class

● **article:** `\part`, `\section`, `\subsection`,
`\subsubsection`, `\paragraph`, `\subparagraph`

● **report, book:** `\part`, `\chapter`, `\section`,
`\subsection`, `\subsubsection`, `\paragraph`,
`\subparagraph`

➤ Commands take title as required argument and short title (for ToC, etc.) as optional argument, e.g.

```
\chapter[Short Title]{Title}
```

➤ Add `*` to command to prevent numbering, e.g.

```
\chapter*[Short Title]{Title}
```



Structure Example

```
\documentclass{article}

\title{My Work} \author{J.~Doe}

\begin{document}

\maketitle

\tableofcontents

\section*{Acknowledgements}

\section{Introduction}

\section[Literature review]{A very long literature
review}
    \subsection{Unsuitable approaches}
        \subsubsection{My comments}

\end{document}
```



Structure Example Result

My Work J. Doe

Contents

Acknowledgements	1
1 Introduction	2
2 Literature review	4
2.1 Unsuitable approaches	4
2.1.1 My comments	8

ACKNOWLEDGEMENTS

1 INTRODUCTION

2 A VERY LONG LITERATURE REVIEW

2.1 Unsuitable approaches

2.1.1 My comments

Itemize

- *Bullet lists* are created by an `itemize` environment

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

- First item.
- Second item.
- Third item.

- *Change labels* on a case-by-case basis

```
\begin{itemize}
\item[$\clubsuit$] clubs
\item[$\heartsuit$] hearts
\end{itemize}
```

- ♣ clubs
- ♥ hearts

Enumerate

➤ *Numbered lists* are generated like itemized lists in an `enumerate` environment

```
\begin{enumerate}
\item Item 1
\begin{enumerate}
\item Item 1.1
\item Item 1.2
\end{enumerate}
\item Item 2
\begin{enumerate}
\item Item 2.1
\end{enumerate}
\end{enumerate}
```

1. Item 1
 - (a) Item 1.1
 - (b) Item 1.2
2. Item 2
 - (a) Item 2.1

Description

➤ For glossaries, etc. use `description` environment

```
\begin{description}
\item[Cabbage] A large
round green vegetable
\item[Brussel sprout] A
small round green
vegetable
\end{description}
```

Cabbage A large round
green vegetable

Brussel sprout A small
round green vegetable



Graphics in L^AT_EX

- `graphics` and `graphicx` packages provide commands to include images
 - In principle any image format (`eps`, `png`, `jpeg`, ...) can be used
 - Which formats can be handled mainly depends on dvi processor
 - PS output: `eps`, (`jpeg`); PDF output: `png`, `jpeg`
- We only discuss the *graphicx* package, `graphics` provides less options

Including Graphics

➤ *Include graphics* file (as box)

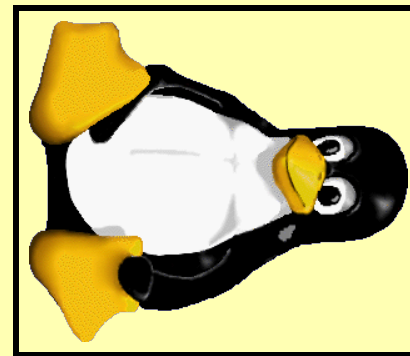
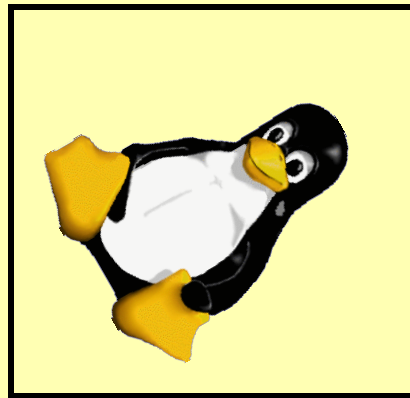
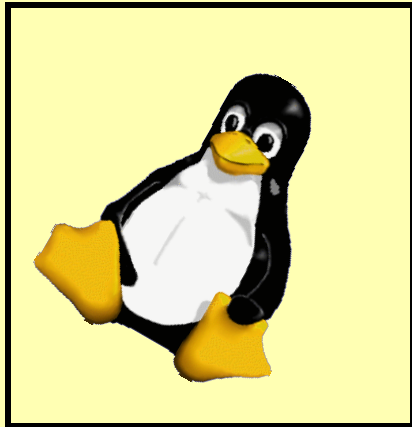
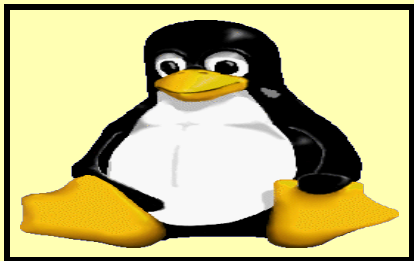
```
\includegraphics [options] {filename}
```

where **options** is a comma separated list of

angle= x	rotate picture by x [°]
width= len	scale picture to width len
height= len	scale picture to height len
scale= x	scale picture
bb= lx by rx ty	set bounding box
clip	clip picture
draft	don't display image, just draw bounding box with filename inside

Graphics Example

```
\includegraphics[height=3cm,width=5cm]{logo.png}  
\includegraphics[angle=-30,width=5cm]{logo.png}  
\includegraphics[angle=-60,width=5cm]{logo.png}  
\includegraphics[angle=-90,width=5cm]{logo.png}
```



Including Tabulars

- *Tabulated material* can be aligned in rows and columns using the `tabular` environment

```
\begin{tabular}{columns} ... \end{tabular}
```

- `columns` is a string describing column types

l left justified column

r right justified column

c centred column

p{l} paragraph column of length l

@{text} column with fixed content text over all rows
(e.g. horizontal space)

*{n}{cols} n copies of cols columns

Row Format

- A row of a `tabular` is *separated into columns by &*
- A *row end* is indicated by `\\`
- Rows may contain less, but not more columns than specified by `tabular` argument

```
\begin{tabular}[t]{p{3cm}lr@{.}l}  
Text in row 1 & Left & 1 & 05\\  
Row 2 & & 341 & 05\\  
& last  
\end{tabular}
```

Text in	Left	1.05
row 1		
Row 2		341.05
	last	

Horizontal and Vertical Lines

- *Vertical lines* are marked by | in column specification
- *Horizontal lines* are inserted with \hline

```
\begin{tabular}{|l|c|}  
\hline  
Item & Cost\\ \hline\hline  
CD & 10.95\\  
Video & 13.20\\ \hline  
\end{tabular}
```

Item	Cost
CD	10.95
Video	13.20

Figures and Tables

- Figures and tables are *floats*
 - They are *float*ed to the nearest convenient location according to some typographical rules
- \LaTeX handles numbering automatically
 - Cross-references can be done using labels
- Figures are created using `figure` environment
- Tables are created using `table` environment
(tables work like figures, but usually have a `tabular` environment instead of `\includegraphics`)

Captions

- A figure or table has a caption and an associated number

```
\caption[short caption]{caption}
```

- **short caption**, if present, is used in list of figures/tables

```
\begin{figure}  
\centerline{\includegraphics[height=4cm]{logo.png}}  
\caption[Tux]{Tux, the Linux Penguin}  
\end{figure}
```

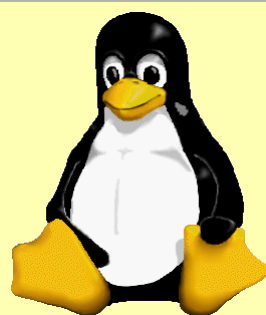


Figure 1.1: Tux, the Linux Penguin

Cross-Referencing

- Assign a *textual label* to last “object”

```
\label{string}
```

- *Objects* are caption, section, subsection, subfigure, ...
(roughly everything that is numbered automatically)

- Refer to the referenced object

```
\ref{string}
```

- Refer to page that object is on

```
\pageref{string}
```

- Cross-referencing requires two T_EX compiler passes
(1st pass detects location and writes it to auxiliary file, 2nd pass inserts it from auxiliary file)

Cross-Referencing Example

```
\section{Introduction}
\label{sec:intro}
\begin{figure} ...
  \caption{Tux}\label{fig:tux}
\end{figure}
See Section~\ref{sec:intro},
Fig.~\ref{fig:tux} on page~\pageref{fig:tux}.
```

1 Introduction

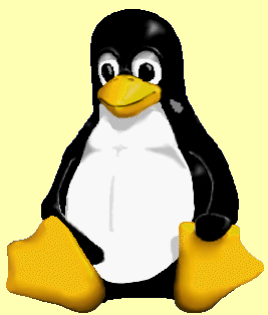


Figure 1.1: Tux

See Section 1, Fig. 1.1 on page 33.

Basic Mathematics

- T_EX/L_AT_EX uses a special mode for typesetting mathematics
- Different versions of formula environments

- *In-line* maths for formula in text

```
\begin{math} ... \end{math}
      \( ... \)          $ ... $
```

- *Display maths* as separated *one-line* formula

```
\begin{displaymath} ...
\end{displaymath}
      \[ ... \]          $$ ... $$
```

- *Numbered one-line display maths* formula

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

Simple Maths Examples

➤ In-line maths

```
Let $x$ be an
integer, s.t.
\ ( x = 2n + 1 \)
```

Let x be an integer, s.t.
 $x = 2n + 1$

➤ Display maths

```
Let
$$f(x) = 4x + 1$$
```

Let
$$f(x) = 4x + 1$$

➤ Numbered display maths

```
\begin{equation}
g(x) = (x-1) / 4
\end{equation}
```

$$g(x) = (x - 1)/4 \quad (1)$$

Subscripts and Superscripts

- Subscripts are created by

```
_ { sub-script }
```

- Superscripts are created by

```
^ { super-script }
```

- Sub-/Super-scripts can be nested and grouped

- Example

```
$$a_l = b_k c^{k_j} +  
d_{x^l} + {f_l}^t$$
```

$$a_l = b_k^l c_j^k + d_{x^l} + f_l^t$$

Fractions and Square Roots

- Fractions are produced using

```
\frac{ numerator }{ denominator }
```

- Roots are produced using

```
\sqrt[ n ]{ formula }
```

- Example

```
$$\frac{\sqrt{2 + z^2}}{\sqrt[3]{a} + 5}$$
```

$$\frac{\sqrt{2 + z^2}}{\sqrt[3]{a} + 5}$$

Greek Letters and Special Functions

- Greek letters are produced by `\` followed by letter name

```

\alpha, \beta, \Gamma, \epsilon, \varepsilon, \tau
\Gamma, \epsilon, \varepsilon, \tau
\varepsilon, \tau

```

- Special functions can be produced using commands like

```
\log, \sin, \exp
```

```

\exp (i\theta) = \cos
\theta + i \sin \theta

```

```
exp(iθ) = cos θ + i sin θ
```

Summations, Products, Limits

➤ Summations and products

```
\sum_{low}^{high}
\prod_{low}^{high}
```

➤ Limits

```
\lim_{limit}
```

```
$$
\lim_{\theta \rightarrow \pi}
\sum_{i=1}^n
\theta^i \sin \theta
$$
```

$$\lim_{\theta \rightarrow \pi} \sum_{i=1}^n \theta^i \sin \theta$$

Multiline Formulæ

➤ Use the `eqnarray` environment for multiple aligned equations

- `eqnarray` numbers each line

```
\begin{eqnarray}
\ln (f(x)) & = & x^2 + \frac{1}{x + 3} \\
f(x) & = & \exp \left( x^2 + \frac{1}{x + 3} \right)
\end{eqnarray}
```

$$\ln(f(x)) = x^2 + \frac{1}{x + 3} \quad (2)$$

$$f(x) = \exp \left(x^2 + \frac{1}{x + 3} \right) \quad (3)$$

Symbols, etc.

➤ Many additional symbols available in maths mode

<code>\approx</code>	\approx	<code>\neq</code>	\neq
<code>\leq</code>	\leq	<code>\geq</code>	\geq
<code>\partial</code>	∂	<code>\pm</code>	\pm
<code>\cdots</code>	\cdots	<code>\vdots</code>	\vdots
<code>\ddots</code>	\ddots	<code>\leftarrow</code>	\leftarrow
<code>\Leftarrow</code>	\Leftarrow	<code>\longleftarrow</code>	\longleftarrow
<code>\Longleftarrow</code>	\Longleftarrow	<code>\rightarrow</code>	\rightarrow
<code>\mapsto</code>	\mapsto	<code>\aleph</code>	\aleph
<code>\forall</code>	\forall	<code>\exists</code>	\exists
<code>\cup</code>	\cup	<code>\cap</code>	\cap
<code>\setminus</code>	\setminus	<code>\times</code>	\times

See literature... and for more also AMS \LaTeX

Bibliographies

- \LaTeX provides a mechanism for citations
 - Symbolic citations to references in a bibliography
 - A special bibliography environment to keep the bibliography itself
- To cite a references in the bibliography

```
\cite [text] {key}
```

- **key** is label of reference
- **text** is additional text printed with reference
- More than one key can be listed, separated by commas

thebibliography Environment

- Reference provided by `thebibliography` environment

```
\begin{thebibliography}
\bibitem[label]{Key} M.~Goossens, F.~Mittelbach.
\emph{The \LaTeX\ companion}.
...
\end{thebibliography}
```

- `key` is used for the `\cite` label
- `label` provides printed label for reference
- Without `label` references are numbered consecutively

- Format of references is free (works like a `itemize` environment)

- BIBTEX generates `thebibliography` automatically
 - Can use large databases containing many references
 - Includes only those that are cited in the document
 - Entries are sorted
 - Entries are consistently formatted (provided the database is consistent)

Bibliography Database

➤ Bibliography databases (.bib) entries

```
@entry-type{key,  
  field1 = "text",  
  ...  
  fieldn = "text"  
}
```

```
@book{eijkhout91,  
  author = "Victor Eijkhout",  
  title = "\TeX\ by Topic, a \TeX{}nicians  
Reference",  
  publisher = "Addison-Wesley",  
  year = "1991"  
}
```

- Many different entry types in bibliography, see literature
- Also many bibtex database tools, e.g. jabref



BIB_TE_X Example

- In \LaTeX source filename.tex with database file database.bib

```
This is the document\dots
```

```
\bibliographystyle{plain} % or unsrt or alpha  
\bibliography{database}
```

- To generate the file

```
> latex filename  
> bibtex filename  
> latex filename  
> latex filename
```