Contents

1 Introduction

1.1 TEX

TEX is a typesetting system which was designed and written by Donald Knuth[1] and first released in 1978. TEX is a popular means of typesetting complex mathematical formulae; it has been noted as one of the most sophisticated digital typographical systems. It was designed with two main goals in mind: to allow anybody to produce high-quality books with minimal effort, and to provide a system that would give exactly the same results on all computers, at any point in time.

1.2 LATEX

LATEX enables authors to typeset and print their work at the highest typographical quality, using a predefined, professional layout. LATEX was originally written by Leslie Lamport. It uses the TEX formatter as its typesetting engine.

In a LATEX environment, LATEX takes the role of the book designer and uses TEX as its typesetter. But LATEX is 'only' a program and therefore needs more guidance. The author has to provide additional information to describe

the logical structure of his work. This information is written into the text as 'LaTeX commands.'

This is quite different from the **WYSIWYG2** (What you see is what you get) approach that most modern word processors, such as MS Word or LibreOffice, take.

1.3 Various Files

1.3.1 Files With LaTeX

.tex LaTeX or TeX input file. Can be compiled with LaTeX .
.sty LaTeX Macro package. Load this into your LaTeX document using the `\usepackage` command.
.dtx Documented TeX . This is the main distribution format for LaTeX style files. If you process a .dtx file you get documented macro code of the LATEX package contained in the .dtx file.
.ins The installer for the files contained in the matching .dtx file. If you download a LaTeX package from the net, you will normally get a .dtx and a .ins file. Run LaTeX on the .ins file to unpack the .dtx file.
.cls Class files define what your document looks like. They are selected with the `\documentclass` command.
.fd Font description file telling LaTeX about new fonts.

1.3.2 Files Outputed

.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.
.aux Another 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.

2 Terminology

2.1 Preamble

The preamble is the first section of an input file, before the text of the document itself, in which you tell LaTeX the type of document, and other information LaTeX will need to format the document correctly.

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\begin{document}
    ...
\end{document}
```

2.2 class files

Contains global processing information for the document. A class file is specified using documentclass command.

2.3 Packages

Packages are a way to extend the functionality of LaTeX . They are loaded using the `\usepackage` command. Common way to use a package is

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

2.4 command

A command begins with a backslash `\` and is a string of characters. A command can have optional and required arguments.

2.5 Environment

Environments are delimited by an opening tag `\begin` and a closing tag `\end`. Everything inside those tags will be formatted in a special manner depending on the type of the environment.

```
\begin{tabular}{ c c c }
    cell1 & cell2 & cell3 \\
    cell4 & cell5 & cell6 \\
    cell7 & cell8 & cell9 \\
   \end{tabular}
```

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

2.6 Document Environment

All that is between `\begin{document}` and `\end{document}`. Can be thought of as "printing" to the document.

2.7 CTAN

Abbreviation for Comprehensive TeX Archive Network. CTAN is a network of servers that hosts the TeX-related software and documentation. It is the central repository for TeX-related software and documentation.

2.8 Latex Distributions

A TeX distribution such is used to produce an output file (such as PDF or DVI) suitable for printing or digital distribution. Some of the popular distributions are:

- TeX Live
- MiKTeX
- MacTeX
- teTex
- proTeXt

2.9 Math Mode

Math mode is a special mode in which you can write mathematical expressions. There are two ways to enter math mode:

- Inline math mode: $a^2 + b^2 = c^2$
- Display math mode:
$$a^2 + b^2 = c^2$$

To enter inline math mode, use a single dollar sign `$`, and to enter Display math mode, use a double dollar sign `$$`.

Apart from this to typeset inline-mode math you can use one of these delimiter pairs: `\( ... \), $ ... $ or \begin{math} ... \end{math}`.

Also, to typeset display-mode math you can use one of these delimiter pairs: `\[ ... \], \begin{displaymath} ... \end{displaymath}`
or
`\begin{equation} ... \end{equation}`.

3 New Environments

3.1 Basic Environment

The new environment definition is achieved by the `\newenvironment` tag:

```
\newenvironment{boxed}
{\begin{center}
\begin{tabular}{|p{0.9\textwidth}|}
\hline\\
}
{
\\\\\hline
\end{tabular}
\end{center}
}
```

Right after the `\newenvironment`, in between braces, you must write the name of the environment, boxed in the example. Below that are two pairs of braces. Inside the first pair of braces is set what your new environment will do before the text within, then inside the second pair of braces declare what your new environment will do after the text.

In the example, in between the before braces a tabular environment is started to draw the vertical lines and a horizontal line is drawn. Inside the after braces another horizontal line is drawn and the tabular environment is closed. Additionally it's enclosed by a center environment.

This is a boxed environment.

3.2 Environments with Parameters

Environments that accept parameters can also be defined.

```
\newenvironment{boxed}[1]
{\begin{center}
#1\\[1ex]
\begin{tabular}{|p{0.9\textwidth}|}
\hline\\
}
{
\\\\\hline
\end{tabular}
\end{center}
}
```

As you see, the command definition is almost the same as in the example of the previous section, except for `[1]` that sets the number of parameters to be used in the environment; and `#1\\[1ex]` that inserts the parameter at the top of the box and also separates the title from the box by a 1ex blank space.

Hello

This is a boxed environment with a title.

3.3 Overwriting existing environments

Environments can be overwritten with `\renewenvironment`. The syntax is equivalent to that of the `\newenvironment` definition.

```
\renewenvironment{itemize}
{\begin{center}\em}
{\end{center}}
```

In this example, the `itemize` environment is overwritten. The new definition is that the `itemize` environment will be enclosed by a center environment and the items will be emphasized.

3.4 Summary of Environments

1. An environment is used with a matching pair of `\begin` and `\end` statements. Both `\begin` and `\end` take the name of the environment as argument in curly braces. The `\begin` statement might have additional mandatory and/or optional arguments.

2. `\newenvironment{<name>}{<begin code>}{<end code>}` defines a new environment called `<name>`. At `\begin{<name>}` the code `<begin code>` will be executed and at `\end{<name>}` the `<end code>` is inserted.

3. `\renewenvironment{<name>}{<begin code>}{<end code>}` can be used to change the existing definition of an environment.