Instructions for use with Acroread

Mouse activated controls:

Quit	quit Acroread
Close	close document
Info	document info
Help	jump to this page
◁—	go back
◁	previous page
Bookmarks	pop-up bookmarks
Thumbnails	pop-up thumbnails
▷	next page
—▷	go forward
internal link	go to link
external link	go to link via browser

Keyboard activated controls:

‘page down’	next page
‘page up’	previous page
‘cursor up’	previous bookmark
‘cursor down’	next bookmark
‘enter’	go to bookmark
‘escape’	window mode

An Introduction to Type-setting projects in LaTeX with the UoYCSProject class

Jeremy Jacob

2007 October 31

What is LaTeX?

LaTeX is a *document description language* built on top of Donald Knuth's TeX type-setting engine.

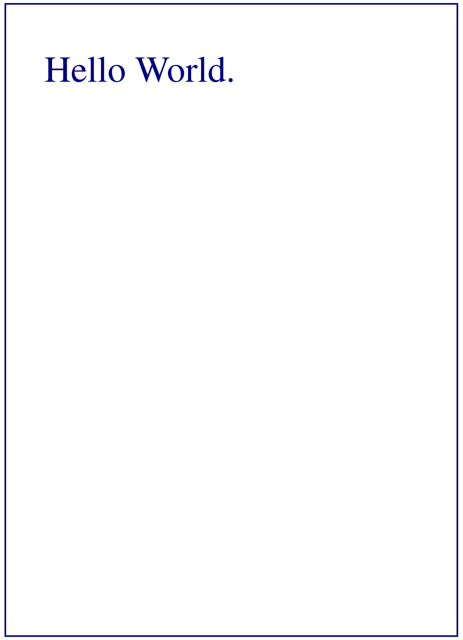
Cf. `HTML` and `SGML/XML` applications.

A minimal document

Source

```
\documentclass{minimal}
\begin{document}
Hello␣World.
\end{document}
```

Output



Hello World.

Why use LaTeX?

- The sophisticated type-setting algorithm of TeX, and the enhanced algorithm of *pdfe(la)tex*. (See the TeX showcase.)
- The huge number of pre-defined packages for doing common things. (See the TeX catalogue.)
- The ability to define your own special purpose structures.
- Stable basis.
- Good for large, academic documents.

References

There are many good references for TEX and friends.

See "*A guide to type-setting project reports in LATEX with the* UoYCSproject class".

UoYCSProject — a class for project reports

There are many pre-defined document classes:

Base minimal, article, report, book, letter, slides.

KOMA-Script scrartcl, scrreprt, scrbook, scrlttr2.

Memorandum memorandum.

Others . . . , beamer, . . . , UoYCSproject, . . .

Text, commands and environments

A LaTeX source is a mix of:

text Some text.,

commands \LaTeX, \frac{2}{3}, and

environments

```
\begin{verse}
  APRIL is the cruellest month, breeding
  \\Lilacs out of the dead land, mixing
  \\...
\end{verse}
```

The anatomy of a LaTeX source

```
\documentclass[class options]{class name}
preamble (definitions and declarations)
\begin{document} % this is a comment.
body
\end{document}
```

The anatomy of a UoYCSproject preamble

```
\documentclass[citation␣style]{UoYCSproject}
%␣Order␣of␣declarations␣does␣not␣matter.
\author{Anne␣Student-Name}
\title{A␣Solution␣to␣the␣Problem␣of␣$\mathit{P}=\mathit{NP}$}
\date{30␣February␣2000}
\supervisor{Prof.␣Z.␣Soporific}
\MEng
\wordcount{2,345}\excludes{Appendix~\ref{sec:code}}
\dedication{To␣My␣Cat,␣Jeoffery}
\abstract{The␣well␣known␣problem␣of␣$P=NP$␣is␣explained,␣together
␣␣with␣its␣significance␣and␣a␣brief␣history␣of␣attempts␣to␣solve
␣␣it.␣␣An␣ingenious␣solution␣is␣presented.}
\begin{document}
...
\end{document}
```

A full list of declarations is given in AGtTSPRiLwtUC.

Extra definitions and package loading

You can load extra packages and make your own definitions.

These go in a file with the same name as your main file, but extension ‘ldf’. **THIS IS DIFFERENT TO THE WAY ALL OTHER CLASSES WORK.** (I have implemented UoYCSproject in this way to ensure that packages are loaded in the correct order.)

Useful packages include: listings , graphics, graphicx, pgf/ tikz , amsmath.

The anatomy of the body

Front matter Title pages, abstract, contents, *&c*.

Main matter The text, divided into (parts,) chapters (, sections, subsections, subsubsections, paragraphs and subparagraphs).

Back matter Bibliography, appendices *&c*.

Front matter

```
\maketitle % Compulsory: title pages, table of contents
\listoffigures % Optional: the list of figures
\listoftables % Optional: the list of tables
... % Optional, package dependent lists ,
....% e.g. \lstlistoflistings
```

Main matter

```
\part{title}          % Optional
\chapter{title}       % Compulsory
\section{title}       % Optional
\subsection{title}    % Optional
\subsubsection{title} % Optional
\paragraph{title}     % Optional
\subparagraph{title}  % Optional
Text. Text.
```

Back matter

```
\bibliography{file1,file2} % Construct bibliography
\appendix % remaining chapters are appendices
\chapter{title}            % One per appendix
\section{title}            % Optional
\subsection{title}         % Optional
\subsubsection{title}      % Optional
\paragraph{title}          % Optional
\subparagraph{title}       % Optional
Text.  Text.
```

Text elements

Characters Can control series, family, shape, colour and size of each text character. See AGtTSPRiLwtUC.

Sentences `Sentence␣one.␣␣Sentence␣two.`

Paragraphs

```
Paragraph␣one.␣␣%␣␣blank␣line␣separates␣paragraphs


Paragraph␣two.
```

Special features

Context dependent emphasis

Cross references Sectional units, floats, equations, *&c.*

Quotations Short and long

Lists Bulleted, numbered and labelled

Tables

Pictures

Floats Tables, Figures and others.

Citations and the bibliography

Through the *natbib* package. Two style options:

Harvard Use option: authoryearcitations

Cite as text 'As Joyce (1939) says, life is complex.'

Cite in parenthesis 'Life is complex (Joyce, 1939).'

Toronto Use option: numericalcitations

Cite as number 'As Joyce [17] says, life is complex.' or 'Life is complex [17].'

Citations are kept in a database in a flat file and processed by a program called BibTEX before inclusion in output file.

Mathematics

Very powerful facilities. May be enhanced by amsmath packages (best advice is to *always* load amsmath).

Inline Here is a formula: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$; isn't it beautiful?

Displayed Here is a formula:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \tag{1}$$

Isn't it beautiful?

Definitions

A major reason for using LaTeX. Create special-purpose commands and environments for the structures in *your* document.

To define a command called \UoY that prints ‘The University of York’:

```
\newcommand∗{\UoY}{The University of York}
```

To define a command that has two parameters:

```
\newcommand∗{\C}[2]{_{#1}C^{#2}}
```

`\begin{math}\C{x+2}{3y}\end{math}` type-sets as ${}_{x+2}C^{3y}$.

Case study: cryptographic protocols — 1

A *message* has three components: *sender*, *receiver* and *content*. So we write our document in terms of a command \msg that has 3 parameters.

Two possible definitions:

```
\newcommand∗{\msg}[3]{#1\rightarrow#2:#3}
\newcommand∗{\msg}[3]{#2\Leftarrow\left[#3\right]\Leftarrow#1}
```

\msg{S}{R}{C^{A^{B}}} produces either $S \rightarrow R : C^{A^{B}}$ or $R \Leftarrow \left[C^{A^{B}}\right] \Leftarrow S$.

Case study: cryptographic protocols — 2a

A protocol is a sequence of messages. So we write our document in terms of an environment that collects a sequence of messages.

We will write, for example:

```
\begin{protocol}
        \msg{A}{B}{X,Y,Z}
  \sep  \msg{B}{C}{W,X}
  \sep  \msg{C}{B}{W,X'}
\end{protocol}
```

Case study: cryptographic protocols — 2b

Now we design the printed form.

The output should have numbered messages to which labels can be attached. Each message should be printed on a line of its own.

The definitions of \msg and \sep should be local to the environment.

Case study: cryptographic protocols — 2c

```
\newcounter{msgnumber}
\newenvironment*{protocol}
 {\setcounter{msgnumber}{0}%
   \newcommand*{\msg}[3]{%
     \refstepcounter{msgnumber}\themsgnumber&##1&##2&##3}
   \newcommand*{\sep}{\\}
   \begin{math}\displaystyle%
     \begin{array}{r@{.\quad}l@{\rightarrow}l@{\;:\;}l}}
 {\end{array}\end{math}}
```

Case study: cryptographic protocols — 2d

Source

```
\begin{protocol}
          \msg{A}{B}{X,Y,Z}
   \sep   \msg{B}{C}{W,X}
   \sep   \msg{C}{B}{W,X'}
\end{protocol}
```

Output

1. $A \longrightarrow B : X, Y, Z$
2. $B \longrightarrow C : W, X$
3. $C \longrightarrow B : W, X'$

How to run LaTeX — 1

1. Create `<source>.tex`, `<source>.ldf`, bibliographic files, *&c.*
2. Run PDF(E)LaTeX 2$_\varepsilon$ (Using TeXLive on Departmental Linux: ‘`pdflatex <source>`’). Collects auxiliary information in `<source>.aux`, `<source>.toc`, *&c.* and creates `<source>.pdf`.
3. Run BibTeX (‘`bibtex <source>`’). This uses the auxiliary information to determine database files and writes `<source>.bbl` file.
4. Run PDF(E)LaTeX 2$_\varepsilon$ (‘`pdflatex <source>`’). Collects auxiliary information in `<source>.aux`, `<source>.toc`, *&c.*, including bibliographic cross references.
5. Run PDF(E)LaTeX 2$_\varepsilon$ (‘`pdflatex <source>`’). There should now be enough auxiliary information to generate the final version of `<source>.pdf`.

How to run LaTeX — 2

Process can be eased by tools such as

- `AUCTeX` package for `emacs` (Linux and Microsoft).
- MikTeX and WinEDT on Microsoft systems.

Incremental processing and errors do not mean repeating the whole process: for example, BibTeX only needs to be re-run if the bibliographic files change or a new citation is added.