

An Introduction to Type-setting projects in \LaTeX 2 ϵ with the **UoYCSProject** class

Jeremy Jacob

The University of York, Department of Computer Science

2009 September 07

What is \LaTeX ?

$\text{\LaTeX} 2_{\epsilon}$ 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 $\text{\LaTeX} 2_{\epsilon}$?

- The sophisticated type-setting algorithm of \TeX , and the enhanced algorithm of *pdf ϵ (l)a*tex. [▶ the \$\text{\TeX}\$ showcase.](#)
- The huge number of pre-defined packages for doing common things. [▶ the \$\text{\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 $\text{T}_{\text{E}}\text{X}$ and friends.
See “*A guide to type-setting project reports in \LaTeX with the UoYCSproject class*” . [▶ Guide](#)

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, scrلتtr2.

Memorandum memorandum.

Others ..., beamer, ..., UoYCSproject [▶ UoYCSproject](#), ...

Text, commands and environments

A $\text{\LaTeX} 2_{\epsilon}$ source is a mix of:

text **Some text.**,

commands **$\backslash\text{LaTeXe}$, $\backslash\text{frac}\{2\}\{3\}$** , and

environments **$\backslash\text{begin}\{\text{verse}\}$**
APRIL is the cruellest month, breeding
 $\backslash\backslash$ Lilacs out of the dead land, mixing
 $\backslash\backslash$...
 $\backslash\text{end}\{\text{verse}\}$

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{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,_Jeoferry}
\abstract{The_well_known_problem_of_{$P=NP$}_is_explained,
  _together_with_its_significance_and_a_brief_history_of
  _attempts_to_solve_it...Aningenious_solution_is_presented.}
\begin{document}
...
\end{document}

```

A full list of declarations is given in [Guide](#), Figure 7.1, P 46.

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 [▶ Guide](#), §6.3.3.

Sentences `Sentence_one. Sentence_two.`

Paragraphs `Paragraph_one. % blank line separates paragraphs`
`Paragraph_two.`

Special features

Context dependent emphasis `\emph{...}\emph{...}...`

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

Quotations Short and long

Citations

Lists Bulleted, numbered and labelled

Tables

Pictures

Floats Tables, Figures and others.

Citations and the bibliography

- Through the `natbib` package, set up for IEEE style.
- `\citep{Joyce:FW}` —cite parenthesised— generates [34], assuming `Joyce:FW` is the label of the 34th reference.
Do not use this form as a noun.
- `\citet{Joyce:FW}` —cite as text— generates Joyce [34], assuming also that the author's surname is 'Joyce'.
You may use this form as a noun.
- Rule: your document should read naturally when all the citation markers (numbers in square brackets, plus the brackets) are removed.

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

► [Example database](#)

Mathematics

Very powerful facilities. May be enhanced by **amsmath** packages (best advice is to *a*lways 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:

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

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  $x+2C^{3y}$ .
```

Case study: Cryptographic protocols

Syntax and form of messages

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

`\newcommand*{\msg}[3]{BODY}`

Two possible definitions for `BODY`:

- ① `#1\rightarrow#2:#3`
- ② `#2\Leftarrow\left[#3\right]\Leftarrow#1`

The call `\msg{S}{R}{C^{A^B}}` produces

- ① $S \rightarrow R : C^{A^B}$ or
- ② $R \Leftarrow [C^{A^B}] \Leftarrow S$ respectively.

Case study: Cryptographic protocols

Syntax of message sequences

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

Desired form of message sequences

Now we design the printed form.

- 1 The output should have numbered messages to which labels can be attached. Each message should be printed on a line of its own.
- 2 The definitions of `\msg` and `\sep` should be local to the environment.

Case study: Cryptographic protocols

Form of message sequences

```

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

```

Case study: Cryptographic protocols

The end product

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

How to run L^AT_EX

The processing cycle

- ❶ Create `<source>.tex`, `<source>.ldf`, bibliographic files, &c.
- ❷ Run PDF(E)L^AT_EX 2_ε (Using T_EXLive on Departmental Linux: `'pdflatex <source>'`). Collects auxiliary information in `<source>.aux`, `<source>.toc`, &c. and creates `<source>.pdf`.
- ❸ Run BibT_EX (`'bibtex <source>'`). This uses the auxiliary information to determine database files and writes `<source>.bbl` file.
- ❹ Run PDF(E)L^AT_EX 2_ε (`'pdflatex <source>'`) a second time. Collects auxiliary information in `<source>.aux`, `<source>.toc`, &c., including bibliographic cross references.
- ❺ Run PDF(E)L^AT_EX 2_ε (`'pdflatex <source>'`) a third time. There should now be enough auxiliary information to generate the final version of `<source>.pdf`.

How to run L^AT_EX

Helpful tools

- Process can be eased by tools such as
 - ▶ AUCT_EX package for **emacs** (any platform).
 - ▶ MacT_EX on Apple
 - ▶ MikT_EX and WinEDT on Microsoft systems.
 - ▶ Eclipse plugin (any platform).
- Incremental processing and errors do not mean repeating the whole process: for example, BibT_EX only needs to be re-run if the bibliographic files change or a new citation is added.
- Most tools also give help with managing BibT_EX databases; there are also many free-standing tools available.