

# 1

## A $\text{\LaTeX}$ quick start

### 1.1 A first example

When you meet a new computer language (and  $\text{\TeX}$  is a language, although not a very general purpose one) the first thing to do is to write a program that outputs **Hello world!** and get it to run. To make  $\text{\LaTeX}$  typeset and print out a message you should perform the following four steps.

(The commands required for steps 2–4 vary from computer to computer. If you are working on a large computer installation then you should ask another user or the system manager what to do. If you are running on a personal computer then look in the manuals for your  $\text{\TeX}$  system or ask the person who installed  $\text{\TeX}$ .)

1. Use a text editor to create a file called `world.tex` containing these lines:

```
\documentstyle{article}
\begin{document}
Hello world!
\end{document}
```

2. Run  $\text{\LaTeX}$  on `world.tex`

When you run  $\text{\LaTeX}$  on `world.tex` you should see messages similar to these appearing on your screen

```
This is TeX, Version 3.0
(WORLD.TEX
LaTeX Version 2.09 <18 March 1992>
(ARTICLE.STY
Standard Document Style 'article'. <14 Jan 92>
(ART10.STY))
No file WORLD.aux.
[1] (WORLD.AUX)
Output written on WORLD.DVI (1 page, 232 bytes).
Transcript written on WORLD.LOG.
```

The messages you see are likely to differ in detail from these. The dates in angle brackets give the release dates of this version of  $\text{\LaTeX}$ . (If the dates you

see are earlier than mid-1991, you should consider updating your system.) The filenames may appear differently depending on the computer you are using. Don't worry about these minor differences, but if you see messages like

```
! Undefined control sequence.
1.1 \documntstyle
      {article}
?
```

or

```
LaTeX error. See LaTeX manual for explanation.
      Type H <return> for immediate help.
! Missing \begin{document}.
?
```

then you have an error. After printing the question mark, T<sub>E</sub>X will stop and wait for instructions. For now just type X in response to the question mark. This causes T<sub>E</sub>X to eXit so that you can re-edit the file `world.tex` and fix your spelling error.

At the end of the run you will find two new files `world.dvi` and `world.log`<sup>1</sup>. The `log` file contains a copy of all the messages that appear on your screen during the run, sometimes with extra information. The `dvi` file is a *device-independent* description of your document that makes no assumptions about the kind of printer you will be using.

### 3. Run a DVI driver

A DVI driver is a program that takes the `.dvi` file and translates it into a printable file for a specific printer. T<sub>E</sub>X uses the graphics capability of the printer to draw the characters rather than simply using the manufacturer supplied characters. There are many different ways of describing graphics commands to printers, and therefore many different DVI drivers are required.

Most DVI drivers have a multitude of options to select which pages are to be printed (useful if you want to check a single change in a large document) and in some cases to print two pages side by side at reduced magnification (and hence reduced quality) on a single sheet, which helps save paper if you just want a draft printout. It would be a good idea to obtain a copy of the manual for your DVI driver. Often it is available as a DVI file, so you can use the driver to print the documentation<sup>2</sup>.

---

<sup>1</sup>On VMS and some other operating systems this file is called `world.lis` to avoid confusion with batch log files

<sup>2</sup>Of course there is a nasty chicken and egg problem here—how will you know how to print the manual without reading it first? Answer: ask someone else who uses your L<sup>A</sup>T<sub>E</sub>X system.

4. Send the output of the driver to the printer. Sometimes the operating system print command is used but usually some special command is required. On some systems, the DVI driver automatically sends its output to the printer, so this step may not be needed at all.

When you have successfully completed these stages you will get something that looks like the example shown on page 4. Just to prove to yourself that there is no relationship between the format of `world.tex` and the resulting output, try editing `world.tex` and putting all the commands on a single line thus:

```
\documentstyle{article}\begin{document}Hello world!\end{document}
```

Run the file through steps 1–4 again and verify that the typeset output is the same.

## 1.2 A large example

The rest of this chapter is a sort of overture to the main book. The idea is to familiarise you with L<sup>A</sup>T<sub>E</sub>X commands by showing you an example document that exercises many L<sup>A</sup>T<sub>E</sub>X features and show you the typeset output and the source file side-by-side. The actual text of the example is meant to be read too — it contains hints on how to avoid simple errors, and when to use some features.

All of the commands in the example are described in detail later in the book. A good way to get started in L<sup>A</sup>T<sub>E</sub>X is to use the structure of this example as a template for your own documents. If you do not understand how a particular effect was created, look at the corresponding part of the source file and then look up those commands in the index to this book.

### 1.2.1 Preliminaries

The paragraphs in a L<sup>A</sup>T<sub>E</sub>X file are separated by blank lines. Most L<sup>A</sup>T<sub>E</sub>X commands are introduced with a `\` character. A few commands generate text directly, such as the command `\today` which will insert the date of the run into your typeset output.

Some commands are *declarations* and do not actually produce any output text, rather they redefine the way the text is to be produced. For instance, the declaration `\bf` causes the following text to be set in a **bold font**.

Curly braces `{` and `}` are used to create *groups*. Any declarations made inside a group will be forgotten at the end of the group, so the commands

```
some {\bf bold} text
produces
some bold text
```

because the effect of the `\bf` declaration is reset at the end of the group.

Many commands take *parameters* which are placed in braces after the command name. For instance the command `\underline{A}` produces A. Some commands take more than one parameter. There are commands that take *optional parameters*

Hello world!

which may be omitted, in which case L<sup>A</sup>T<sub>E</sub>X will take some default action. Optional parameters are marked with brackets [ and ] instead of braces.

This book uses a special font to typeset fragments of L<sup>A</sup>T<sub>E</sub>X code.

`Text like this` represents L<sup>A</sup>T<sub>E</sub>X instructions that you can type in directly.

*Text in italics* represents a L<sup>A</sup>T<sub>E</sub>X part-of-speech that you must replace with a specific name or piece of text.

For instance, elsewhere in this book you will be told that the command

```
\underline{text}
```

causes *text* to be typeset with underlining. You can type anything you like (including other L<sup>A</sup>T<sub>E</sub>X commands<sup>3</sup>) instead of *text*, but you must type the `\underline` and the braces as shown.

Commands of the form

```
\begin{environment name} ... \end{environment name}
```

define the start and end of *environments* and must be paired. Within an environment some special layout will be used to typeset your text. For instance, any text between a `\begin{center}` and an `\end{center}` is centred on the page. (Note the American spelling of `center` here.) Any declarations made within an environment will be forgotten after the corresponding `\end` command, just as they would be at the end of a group. Groups and environments may be nested.

All L<sup>A</sup>T<sub>E</sub>X document files must have the same basic form as the `world.tex` file. There must be exactly one `\documentstyle{style}`, one `\begin{document}` and one `\end{document}` command in that order. The part of the document file before the `\begin{document}` is called the *preamble*. No text is produced by the preamble, and blank lines are ignored. Text generating commands are in fact illegal in the preamble, which must include only declarations, such as the title of the document. Some declarations may *only* appear in the preamble. Many documents (such as `world.tex`) have empty preambles because the default formatting is acceptable.

### 1.3 Going further

In the Spring term I will give a lecture which discusses L<sup>A</sup>T<sub>E</sub>X font selection and the formatting of maths and tables. A later lecture will describe L<sup>A</sup>T<sub>E</sub>X page layout and style file construction. If you are in a hurry, you could look at a copy of my book L<sup>A</sup>T<sub>E</sub>X, *concisely*, or perhaps try one of the other three L<sup>A</sup>T<sub>E</sub>X books that are available. I have copies of all four books and you are welcome to come and look through them, but I'm afraid I don't lend books out anymore after some previous bad experiences...

---

<sup>3</sup>Some commands may place restrictions on what other commands you may use inside their parameters.

# A L<sup>A</sup>T<sub>E</sub>X document

Adrian Johnstone\*

June 1992

## Abstract

This document contains examples of many L<sup>A</sup>T<sub>E</sub>X features. It is taken from Chapter 1 of ‘L<sup>A</sup>T<sub>E</sub>X, *concisely*’.

## 1 Introduction

T<sub>E</sub>X is a computer program for typesetting papers and books. L<sup>A</sup>T<sub>E</sub>X is a package of *macros* that add features such as cross referencing, table of contents generation and automatic compilation of bibliographies. It is designed to feel rather like systems such as the Unix<sup>1</sup> **troff** and VMS<sup>2</sup> Runoff packages.

Auxiliary programs help in making sorted bibliographies, indices and glossaries. You should also find out if a spelling checker is available on your computer.

### 1.1 A subsection

L<sup>A</sup>T<sub>E</sub>X provides *sectioning* commands for parts, chapters, sections, subsections, subsubsections and others. It will automatically keep track of section numbers and generate a table of contents for you.

---

Computer Science Department, Royal Holloway, University of London

<sup>1</sup>Unix is a trademark of AT&T

<sup>2</sup>VMS is a trademark of Digital Equipment Co.

```

\documentstyle{article}

%This is a comment.
%A comment is everything after a % sign up to the end of line.

\setlength{\textheight}{160mm}    %To fit Ellis Horwood
\setlength{\textwidth}{115mm}     %To fit Ellis Horwood

\title{A \LaTeX\ document} \author{Adrian Johnstone% ignore eol
\thanks{Computer Science Department, Royal Holloway, University
of London}} \date{June 1992}

% End of preamble

\begin{document} %Start of real document
\maketitle

\begin{abstract}This document contains examples of many \LaTeX\
features. It is taken from Chapter 1 of '\LaTeX, {\em
concisely}'.\end{abstract}

\section{Introduction}

\TeX\ is a computer program for typesetting papers and books.
\LaTeX\ is a package of {\em macros} that add features such as
cross referencing, table of contents generation and automatic
compilation of bibliographies. It is designed to feel rather like
systems such as the Unix\footnote{Unix is a trademark of AT\&T}
{\tt troff} and VMS\footnote{VMS is a trademark of Digital
Equipment Co.} Runoff packages.

Auxiliary programs help in making sorted bibliographies, indices
and glossaries. You should also find out if a spelling checker is
available on your computerr.

\subsection{A subsection} \LaTeX\ provides {\em sectioning}
commands for parts, chapters, sections, subsections,
subsubsections and others. It will automatically keep track of
section numbers and generate a table of contents for you.

```

## An unnumbered subsection

If you follow the sectioning command with a \* then the number is suppressed, and no table of contents entry is generated.

Many commands have \*-forms that slightly modify their behaviour.

## 2 Things to watch when you are typing

Usually a L<sup>A</sup>T<sub>E</sub>X document will mostly be straight text with only occasional embedded formatting commands (this document has a very high proportion of formatting commands because it is designed to show off many features). However, even in straight text there are certain things you should watch for because typeset text is not the same as normal computer printout.

- words are separated by one or more spaces, but L<sup>A</sup>T<sub>E</sub>X makes its own decisions as to how to space the output. So it doesn't matter how many spaces you type.
- paragraphs are separated by one or more blank lines.
- don't use the keyboard double quote character ". Your keyboard has single left quote ' and single right quote ' keys. Reported speech is usually surrounded by double quote marks "thus". Reported speech within reported speech, and the first novel use of a technical term is usually surrounded by single quotes: "He said 'don't quote me on that'".
- T<sub>E</sub>X breaks lines by looking for interword spaces and sometimes by hyphenating. You can make an unbreakable interword space with a *tie* which is written ~ ensuring that the Mr in Mr Smith is never orphaned. You can disable line breaking altogether (including hyphenation) by putting text in a box which will simply hang out into the margin.
- You have already seen examples of *emphasized words* typeset in italic. *Emphasis within emphasis is shown in Roman type. But don't overdo it.*<sup>1</sup>

---

<sup>1</sup>This is a footnote to the emphasized text. Note the use of an italic correction to place the footnote mark correctly.

`\subsection*`{An unnumbered subsection} If you follow the sectioning command with a `{\tt *}` then the number is suppressed, and no table of contents entry is generated.

Many commands have `{\tt *-}`forms that slightly modify their behaviour.

`\section`{Things to watch when you are typing} Usually a `\LaTeX` document will mostly be straight text with only occasional embedded formatting commands (this document has a very high proportion of formatting commands because it is designed to show off many features). However, even in straight text there are certain things you should watch for because typeset text is not the same as normal computer printout.

`\begin{itemize}`

`\item` words are separated by one or more spaces, but `\LaTeX` makes its own decisions as to how to space the output. So it doesn't matter how many spaces you type.

`\item` paragraphs are separated by one or more blank lines.

`\item` don't use the keyboard double quote character `{\tt "}`. Your keyboard has single left quote `{\tt '}` and single right quote `{\tt '}` keys. Reported speech is usually surrounded by double quote marks "thus". Reported speech within reported speech, and the first novel use of a technical term is usually surrounded by single quotes: "He said 'don't quote me on that',".

`\item` `\TeX` breaks lines by looking for interword spaces and sometimes by hyphenating. You can make an unbreakable interword space with a `{\em tie}` which is written `{\tt ~{}}` ensuring that the Mr in Mr~Smith is never orphaned. You can disable line breaking altogether (including hyphenation) by putting text in a box which `\mbox`{will simply hang out into the margin.}

`\item` You have already seen examples of `{\em emphasized words}` typeset in italic. `{\em Emphasis {em within} emphasis` is shown in `{\em Roman}` type. But don't overdo it. `\footnote`{This is a footnote to the emphasized text. Note the use of an italic correction to place the footnote mark correctly.}

`\end{itemize}`

- although your keyboard probably has only one kind of dash, typeset text requires four different kinds: the intra-word hyphen; a dash for numeric ranges (17–25); a punctuation dash—which should not have spaces round it—and the mathematical minus sign (17 – 25).
- these keyboard characters are already reserved by L<sup>A</sup>T<sub>E</sub>X and need special treatment

# \$ % & \ { } - ^ ~

### 3 Displays

The bulleted list in the previous section is one kind of *display*. You can make numbered lists

1. first item
  - and you can nest lists of different kinds
  - ◊ as well as override the default tick mark
2. second item

Quotations may also be displayed by using indented paragraphs

‘Paragraph indentation is suppressed for short quotations’

There is another quotation environment for multiparagraph quotes.

Longer quotations (those with more than one paragraph) use extra indentation ...  
...at the start of paragraphs.

### 4 Type styles and sizes

By default, L<sup>A</sup>T<sub>E</sub>X sets type in a font called ‘roman’. There are seven type styles that may be used with ordinary text—roman, **bold**, sans serif, *slanted*, SMALL CAPS and **typewriter**.

You can have tiny, scriptsize, footnotesize, small, normalsize, large, **Large**, **LARGE**, **huge** and **Huge** text.

```

\begin{itemize}
\item although your keyboard probably has only one kind of dash,
typeset text requires four different kinds: the intra-word
hyphen; a dash for numeric ranges (17--25); a punctuation
dash---which should not have spaces round it---and the
mathematical minus sign ( $17-25$ ).

\item these keyboard characters are already reserved by \LaTeX\
and need special treatment

\begin{center}
\# \$ \% \& \$\backslash$ \{ \} \_ \^ \~\
\end{center}
\end{itemize}

\section{Displays}

The bulleted list in the previous section is one kind of {\em
display}. You can make numbered lists \begin{enumerate} \item
first item \begin{itemize} \item and you can nest lists of
different kinds \item[ $\diamond$ ] as well as override the default
tick mark \end{itemize} \item second item \end{enumerate}
Quotations may also be displayed by using indented paragraphs
\begin{quote} 'Paragraph indentation is suppressed for short
quotations' \end{quote} There is another quotation environment
for multiparagraph quotes. \begin{quotation} Longer quotations
(those with more than one paragraph) use extra indentation \ldots

\ldots at the start of paragraphs.

\end{quotation}
\section{Type styles and sizes}

By default, \LaTeX\ sets type in a font called 'roman'. There are
seven type styles that may be used with ordinary text---roman,
{\bf bold}, {\sf sans serif}, {\sl slanted}, {\sc small caps} and
{\tt typewriter}.

You can have {\tiny tiny}, {\scriptsize scriptsize},
{\footnotesize footnotesize}, {\small small}, normalsize, {\large
large}, {\Large Large}, {\LARGE LARGE}, {\huge huge} and {\Huge
Huge} text.

```

## 5 Mathematics

Mathematical formulae can be in-text like  $x = \sum_{i=1}^{15} y_i/y_{i+1}$  or displayed

$$x = \sum_{i=1}^{15} \frac{y_i}{y_{i+1}}$$

and the position of subscripts and superscripts will be tweaked as necessary. This example shows the use of subscripts, superscripts and fractions. There are other commands for building matrices and large brackets.

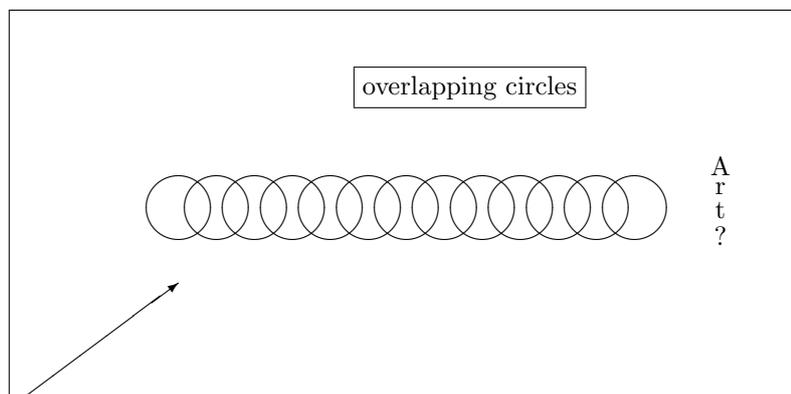
$\TeX$  knows about dozens of maths symbols. Here are just a few

$\aleph$	$\Re$	$\Im$	$\cup$	$\hbar$	$\wp$	$\iota$	$j$	$\ell$	$\nabla$	$\partial$	$\surd$	$\forall$	$\exists$	$\neg$	$\setminus$
$\top$	$\perp$	$\parallel$	$\angle$	$\clubsuit$	$\diamond$	$\heartsuit$	$\spadesuit$	$\prime$	$\emptyset$	$\infty$	$ $	$\flat$	$\natural$	$\#$	

## 6 Figures, tables and pictures

$\LaTeX$  has an easy to use table construction command. The table of maths commands above is an example of its use. You can make tables *float* which means that they will move to the top or bottom of a page of a page. The `table` and `figure` environments make floats and allow captions to be entered automatically into a list of figures or tables.

Finally, you can draw simple pictures like this



```
\section{Mathematics}
```

Mathematical formulae can be in-text like  $x=\sum_{i=1}^{15} y_i/y_{i+1}$  or displayed 
$$x=\sum_{i=1}^{15} \frac{y_i}{y_{i+1}}$$
 and the position of subscripts and superscripts will be tweaked as necessary. This example shows the use of subscripts, superscripts and fractions. There are other commands for building matrices and large brackets.

$\TeX$  knows about dozens of maths symbols. Here are just a few

```
\begin{center}\begin{tabular}{*{16}{c}}
```

```

 $\aleph$  &  $\Re$  &  $\Im$  &  $\mho$  &  $\hbar$  &  $\wp$  &  $\imath$  &
 $\jmath$  &  $\ell$  &  $\nabla$  &  $\partial$  &  $\surd$  &  $\forall$ 
&  $\exists$  &  $\neg$  &  $\backslash$ 
 $\top$  &  $\bot$  &  $\|$  &  $\angle$  &  $\clubsuit$  &  $\diamondsuit$ 
&  $\heartsuit$  &  $\spadesuit$  &  $\prime$  &  $\emptyset$  &
 $\infty$  &  $\vert$  &  $\flat$  &  $\natural$  &  $\sharp$ 

```

```
\end{tabular}\end{center}
```

```
\section{Figures, tables and pictures}
```

$\LaTeX$  has an easy to use table construction command. The table of maths commands above is an example of its use. You can make tables `{\em float}` which means that they will move to the top or bottom of a page of a page. The `{\tt table}` and `{\tt figure}` environments make floats and allow captions to be entered automatically into a list of figures or tables.

Finally, you can draw simple pictures like this

```
\vspace*{1ex}
\begin{center}
\setlength{\unitlength}{1mm}
\fbbox{
\begin{picture}(100,50)
\multiput(20,25)(5,0){13}{\circle{8}}
\put(43,40){\framebox{overlapping circles}}
\put(90,20){\shortstack{A\r\t\?}}
\put(0,0){\vector(4,3){20}}
\end{picture}
}
\end{center}
```