Tutorial 3: Organising and adding style in LaTeX 2ε

Pierre Chardaire ©UEA

July 2, 2017

> *Enlightenment must come little by little, otherwise it would overwhelm.*
>
> Idries Shah

In this tutorial you will learn how to

- organise your document,
- add style to your paragraphs,
- and create lists.

1 Organizing a document

Begin a new document and type in the following:

```
% Document for tutorial 3
\documentclass[12pt]{scrartcl}
\title{The life of Brian}
\author{John Doe
        \thanks{Third year CMP student}}
\begin{document}
\maketitle
\end{document}
```

Save the document and process it as usual. Add an abstract by including the following just after `\maketitle`:

```
\begin{abstract}
The Life of Brian is not really interesting but I am
paid to do this job. No one would remember Brian’s
life if not for a mad lecturer who used it as an
illustrative example in a tutorial at the University
of East Anglia.
\end{abstract}
```

The abstract starts with `\begin{abstract}` and ends with `\end{abstract}`. Such delimited blocks are called *environments*. Besides the *abstract environment* you have already seen the *document environment*, and you’ll see more soon.

Warning: I know you are bit lazy (☺) but if you cut and past from this PDF (your PDF external viewer should allow it) you could have a problem. Try it with the abstract above and process your document. If the compilation succeeds read carefully the abstract in the produced PDF. Do you see the problem? Now, look carefully at your tex file. You should notice that the input apostrophe in “Brian’s life” is not the right ASCII character. Correct and reprocess[1].

Various levels of sectioning can be added to your document by using the `\section`, `\subsection` and `\subsubsection` commands as follows:

```
\section{Introduction}
Everyone knows, even the dumbest reader, that Brian was
an ordinary bloke.

\section{Infancy}
As Albert Einstein, Brian was a late developer, except
that, contrary to Einstein, Brian never developed very
much.
```

[1] More generally your LaTeX source must only contain ASCII characters. Anything you copy from Word or other similar tool will be from an extended set of characters. Then pasting to the LaTeX editor will cause problems that may be difficult to identify. For example, a non-ASCII character may appear as a blank in your text but could cause a strange error when processing with LaTeX. You’ve been warned! Why copy from Word anyway?

```
\subsection{Before the age of six}

Absolutely nothing happened to Brian until the age of
six\ldots{} and not much after.

\subsubsection{Do we need to elaborate?}
No!

\section{Acknowledgement}
The help of the Brian Trust was unvaluable in writing
this paper.
```

Process and examine the result. In general the acknowledgement section is not numbered. Replace the command `\section{Acknowledgement}` with the starred version `\section*{Acknowledgement}`; recompile and examine the result.

You can also include footnotes:

```
As Albert Einstein, Brian was a late developer, except
that, contrary to Einstein, Brian never developed very
much\footnote{According to his tutor Albert Einstein.}.
```

Notice that the full stop of the sentence must come after the footnote.

2 Stylistic elements

You can emphasize some of the text using the `\emph` command:

```
As Albert Einstein, Brian was a late developer, except
that, contrary to Einstein, Brian \emph{never} developed
very much\footnote{According to his tutor Albert
Einstein.}.
```

You can use various font styles. You may sometimes want to write in bold, for example

```
\textbf{Absolutely nothing happened} to Brian until the
age of six\ldots{} and not much after.
```

and you can even emphasize within the bold section as follows:

```
\textbf{Absolutely \emph{nothing} happened} to Brian
until the age of six\ldots{} and not much after.
```

Some of you may want to describe computer language terms/sentences. This is commonly done by using a typewriter font, e.g.

```
In \texttt{C++} you print ``hello world''
by using the statement.
\begin{center}
\texttt{cout << "hello world" << endl;}
\end{center}
```

Note the effect of the *center environment*. It starts a new line and centres everything it includes. Notice also that the traditional non-matching double quote character is used to get the desired effect in the printed `C++` code.

3 Using lists

You can use a number of environments to create lists.

- Bullet point lists (like this one) use the *itemize environment*.
- Numbered lists use the *enumerate environment*.
- Lists that are indexed by words use the *description environment* .

Here is an example of a bullet point list:

```
You can use a number of environments to create
lists.
\begin{itemize}
\item Bullet point lists (like this one)
      use the \emph{itemize environment}.
\item Numbered lists use the
      \emph{enumerate environment}.
\item Lists that are indexed by words use
      the \emph{description environment}.
\end{itemize}
```

So you start with `\begin{itemize}` and end with `\end{itemize}`. Between the two, every `\item` corresponds to a bullet point.

Now, an example of an enumerated list:

```
My program for today:
\begin{enumerate}
\item Finish this tutorial on \LaTeX.
\item Have a nap.
\item Start another tutorial.
\end{enumerate}
```

This is very similar to the previous environment except that in this case you start with `\begin{enumerate}` and end with `\end{enumerate}`. Between the two, every `\item` corresponds to a numbered point.

You can also write description lists as follows:

```
Computer languages are classified as follows:
\begin{description}
\item[C] a language for nerds
\item[Java] a language for people who do not understand
pointers
```

```
\item[C++] a language for super-nerds
\item[Prolog] a language for aliens
\end{description}
```

You start with `\begin{description}` and end with `\end{description}`. Between the two, every `\item` takes a parameter between square brackets which is a "name" for the item described.

You can have lists within lists and mix the types of list:

```
Difference between languages:
\begin{itemize}
\item C is a \emph{procedural} language:

  \begin{itemize}
  \item This is a low level language used to implement
  OS.
  \item The programmer has to manage the dynamic memory.
  \end{itemize}

\item Java is an \emph{object oriented} language:

  \begin{enumerate}
  \item Pointers are hidden to the user.
  \item Dynamic memory is managed by the system.
  \end{enumerate}

\end{itemize}
```

Note the blank lines do nothing. (Try to suppress them and reprocess.) The formatting of the input just adds clarity to your tex file.

4 Other useful environments

There are other environments that you may find occasionally useful.

For example try to process the following:

```
There is a lot of wisdom in popular sayings about
the weather.
You probably know this famous proverb:
\begin{quote}
Red sky at night, shepherd’s delight.\\
Red sky in morning, fisherman’s warning.
\end{quote}
but did you know this French one:
\begin{quote}
Rain in November, Christmas in December.
\end{quote}
```

The most important environments are presented in the *The Not So Short Introduction to LaTeX2e*, but there are many others. No need to learn what you’ll seldom need. Just look it up in your LaTeX manual or Google it when the occasion arise.

Now, have a little nap to rest your tired brain … More to come!