MAS116/117 Presentation Lecture 3: Packages, Environments and More Presentation

Dr Simon Willerton

The name of the game

> *"English words like 'technology' stem from a Greek root beginning with the letters τέχ...; and this same Greek word means* art *as well as technology. Hence the name TEX [...] TEX rhymes with the word blecchhh" [Donald Knuth 1984]*

Leslie Lamport built LaTeX on top of TeX and added part of his name to it. It is typically pronounced 'lay-tek'.

1 Packages

Loading packages

Often computer languages such as Python and LaTeX have optional features that can be included.

- In Python you 'import modules'.
- In LaTeX you 'use packages'.

In LaTeX you put `\usepackage{..}` in the preamble.

For example we have already seen the `parskip` package.

```
\documentclass[a4paper, 11pt]{amsart}

\usepackage{parskip}

\begin{document}
```

There are thousands of LaTeX packages!

We will see a few of them.

AMS packages I

Some important mathematical features are provided by packages from the American Mathematical Society (AMS).

For example, if you want to use the symbol $\mathbb{N}$ for the natural numbers, then you want to use a 'blackboard bold' font.

This is provided by the `amssymb` package, so you need to included in the preamble.

```
\usepackage{amssymb}
```

Then to get "Suppose $x \in \mathbb{N}$." you can write

```
Suppose $x \in \mathbb{N}$.
```

You should include `amssymb` in any mathematical document.

AMS packages II

Two other packages from the AMS are `amsmath` and `amsthm`.

- `amsmath`: provides many, many useful things you will use, e.g. the `align*` environment for aligning '=' signs.
- `amsthm`: provides 'theorem-like' environments which you will see in the lab.

[Note, `amsmath` is loaded automatically by `amsart` but you should always include it if you use the `article` class.]

Having mentioned 'environments' I should say what they are...

2 Displaying computer code using environments

Marking up with environments

There are three basic ways to 'mark up' text in LaTeX.

- By putting it inside `$..$` or `\[..\]`.
- By putting it inside `{..}` following a command, e.g. `\emph{hello}` or `\section{Introdution}`.
- By putting it inside an environment,

```
\begin{itemize}
    \item One thing.
    \item Another thing.
```

```
\end{itemize}
```

Environments begin with `\begin{..}` and end with `\end{..}`.

I indent environments with four spaces to make my file more readable. (But as we will see this is not always possible.)

Entering computer code

Entering computer code straight into a LaTeX document can cause problems due to the special characters used in programming. For example, showing LaTeX commands in a LaTeX document can't be achieved by simply typing them. (Why?)

The `verbatim` environment solves this problem, and is useful for entering small bits of computer code.

The verbatim package

For example, to get the output

```
Here's the \emph{emphasise} command.
```

you can use

```
\begin {verbatim}
Here's the \emph{emphasise} command.
\end {verbatim}
```

If you didn't put it in the `verbatim` environment then you'd get

Here's the *emphasise* command.

If you indent here, the spaces will appear in the output.

The listings package

For long passages of computer code there are better options than verbatim.

The `listings` package has the `lstlisting` environment, which allows long scripts to appear with line-numbers, commands highlighted and more.

```
1 # A simple python script
2
3 name = input("What is your name? ")
4
5 for i in range(1, 10):
6     print("Hello", name)
7
8 if name == "Simon":
```

```
9       print("That's my name too!")
10  else:
11      print("Nice to meet you!")
```

The Python lab sheets use the `minted` package to colour the code. However, this requires more software to be installed on your machine.

3 Activity time

Activity. The circulated document was created with a fairly minimal LaTeX file. Can you work out which commands were used where?

On a print-out of the document (or a separate piece of paper), recreate the main LaTeX commands as closely as possible, looking out for environments in particular.

4 More on presentation

Thinking about presentation

I hope you are starting to think more about how you write mathematics than you have before. This should not only happen when doing homeworks for this module, but also homeworks for other modules. For example, you should have now received marked homework from some of those other modules. Did you get comments there relating to how you presented your ideas?

Typical feedback

You may have received comments about the following:

- not using full sentences
  (use more words to make the writing flow);
- being imprecise with notation
  (try hard to learn from your lecturer or tutor's solutions);
- using 'woolly' arguments
  (can you make things more precise?);
- 'proving by example'
  (you can only *disprove* by (counter-)example!).

Overall, presenting maths well does take effort but the effort will pay off (in better understanding and higher marks).

5 In this week's lab...

About Computer Lab 3

In Computer Lab 3 we'll look at using some of LaTeX's auto-referencing features, looking at how to create numbered equations, propositions and theorems, and how to refer to them later.

$$E = mc^2 \tag{1}$$

Proposition 5.1. *Every even number is the sum of two odd numbers.*
Theorem 5.2. *There are an infinite number of prime numbers.*

Proposition 5.1 is easier to prove than Theorem 5.2.