This lesson shows how you can extend LaTeX to your needs and change its layout further by using packages and definitions. It also shows how you can define your own commands.

After having declared a class, in the preamble you can modify functionality in LaTeX by adding one or more packages. These can

- Change how some parts of LaTeX work

- Add new commands to LaTeX

- Change document design

1 Changing how LaTeX works

`babel`   is to change how LaTeX deals with language-specific typesetting (hyphenation, punctuation, quotations, localisation, etc.). Different languages have different rules, so it's important to tell LaTeX which one to use.

This is a lot of filler which is going to demonstrate how LaTeX hyphenates material, and which will be able to give us at least one hyphenation point. This is a lot of filler which is going to demonstrate how LaTeX hyphenates material, and which will be able to give us at least one hyphenation point.

2 Changing design

`geometry`   It's useful to be able to adjust some aspects of design independent of the document class. The most obvious one are the page margins. For that we use `geometry`.

3 Defining commands

Sometimes you need a command specific to your document, either some functionality not found in the available packages or simply a command to enter a common expression that is used multiple times.

```
\newcommand\kw[1]{\textbf{\itshape #1}}
```

In the definition [1] denotes the number of arguments (here one) and #1 denotes the first argument that is supplied (apples or oranges in this example). You may have up to nine arguments, but it is usually best to have just one argument, or sometimes none at all.

Usage   I like **cats**. I like *cats*.

4 Exercises

4.1 Generate code in green

Use the command below

```
\newcommand{\greencode}[1]{\textcolor{teal}{\texttt{#1}}}
```

Usage   `int x = 3;`

4.2 Creating a command without arguments

Use the command below

```
\newcommand{\printerror}{\textcolor{red}{ERROR}}
```

Usage   ERROR