Commenting

Comments are used to add notes into your program. There are several reasons to do this.

Comments in Python start with a # and continue to the end of the line. They can take a whole line:

```
# This is a comment which takes the whole line.
```

or they can be put at the end of a command

```
print("Hello World!") # This comment starts after the print command.
```

You can use several comment lines for a long comment.

```
# Here is a comment that is quite long and needs to spread over
# multiple lines in order to say what needs to be said.
```

You can put anything in a comment, so sometimes you see a block of special characters designed to draw the eye of a person (yourself or someone else) who is scanning quickly through the code by eye looking for something.

```
#~#~#~#~#~#~#~#~#~#~@~@~#~#~#~#~#~#~#~#~#
#~#~#~#~#~#~#~@~@~@~@~@~@~@~@~#~#~#~#~#~#
#~#~#~#~@~@~@~@~@~@~@~@~@~@~@~@~#~#~#~#
#~#~@~@~ THIS BIT IS IMPORTANT ~@~@~#~#
#~#~#~#~@~@~@~@~@~@~@~@~@~@~@~@~#~#~#~#
#~#~#~#~#~#~@~@~@~@~@~@~@~#~#~#~#~#~#
#~#~#~#~#~#~#~#~#~#~@~@~#~#~#~#~#~#~#~#~#
```

Why comment?

Explaining your code

It is a good idea to write comments to explain what your code is doing. These might explain your code so someone else can see what you have done and how (for example, me when marking your coursework).

You don't just comment code that is written for others, though. If you look at a program you wrote six months, a year, five years ago, you might find it takes you time to work out what the code you wrote is doing. Comments are useful notes to your future self that act as a guide to what a program is doing.

Sometimes a comment at the end of a command is useful to make a note about what the command is doing. Sometimes a whole line comment is useful to say what the next bit of code is going to do.

Your code comments should be minimal and should only appear where there is a need to explain what is happening to an experienced programmer because it isn't obvious from looking at the code.

For example, the following comment is totally unnecessary:

```
for i in range(0,10): # loops ten times with i=0,..9
```

The code already tells an experienced programmer what the comment says, so it shouldn't be here.

You'll see some of the code snippets in the examples in these notes contain comments. Usually these are over-the-top for what you would write in your comments. This is because I am writing comments to try to teach you how the code works, whereas you are writing comments so that an experienced programmer will be able to work out what your code is doing. Please try to understand this difference when writing your own comments. Don't worry though, just try something and I'll tell you in the weekly work if you are writing too much.

Placeholder when writing code

Sometimes it is useful to plan out an algorithm before implementing it in code. Then you might leave comments to remind yourself what bits go where.

```
for i in range(0,100):
    # TO DO: code here checks whether i is prime and if so prints it
    print(i) # for now, just print all values of i
```

When testing or debugging code

Commenting can also be useful when debugging or checking a program works as expected. For example, if a line of code generates a random number like this:

```
from random import uniform
x=uniform(1, 10)
```

Say you are worried about the case where x equals 5. Rather than re-running the code until x happened to be 5, you could simply comment out the uniform command and temporarily replace it with $x=5$. You would comment out the command rather than deleting it because once you are sure the code that follows is handling $x=5$ correctly, you can restore the line by removing the # and deleting or commenting out the $x=5$ line.

```
from random import uniform
#x=uniform(1, 10)
x=5
```

Quick shortcut for commenting

Often editors have a command that adds a comment, so you can highlight several lines of code and use this to add a # to all of them. For example, in IDLE on my computer there is an option in the 'Format' menu called 'Comment Out Region' which does this and has the shortcut Ctrl-D. So I can highlight a block of code I don't want to run and press Ctrl-D to comment it out, and later press Ctrl-Shift-D to uncomment it again.