

ECMM455 Python Worksheet 1: Getting started with Python

Prof Hywel Williams

September 12, 2019

1 Aims

- Learn to use Python with the Spyder IDE
- Create and run your first Python program

2 This document

This document is the first of many worksheets that will guide you through learning to program in Python. Python commands are *highlighted* in the text. Examples of Python output are given in boxes like this:

```
>>> print("this is some sample output")  
  
this is some sample output  
  
>>>
```

Examples of Python code and programs are given in boxes like this:

```
1 print("this is some example code")  
2  
3 x = 3*y
```

At times there may be a few visual differences between the examples in the worksheet and what you see on the screen - I use the Linux Ubuntu operating system, whereas you may be using a different operating system. Make sure you read through the examples and understand what they do. Repeat them in Python to check you get the same answers!

This document is not sufficient for you become an expert in Python. This will require that you become familiar with the sources of online help and invest time in practicing. Like all computer programming, competence in Python is a skill to be developed - not a set of facts to be learned - and will only be gained by experience.

3 Getting started

3.1 Python version?

In this course we will learn to use Python 3.7, the version shipped as standard with the latest version of Anaconda.

3.2 Launch Spyder

Spyder is an integrated development environment (IDE) which provides a number of useful features for Python programming, primarily an editor (in which you can write code) and two forms of console (in which you can execute single commands or entire programs). Documentation is available here: <https://docs.spyder-ide.org/>

To get started, launch Spyder (how you do this will depend on your OS and Anaconda install). Then take a quick look at the built-in tutorial (find it in the Help menu). This will give you some pointers as to what features are available in Spyder and how you can run a program you have written in the editor.

3.3 Using interactive mode

Interactive mode allows you to type commands directly into the Python interpreter. Output is then presented immediately. Interactive mode is useful for performing small tasks (anything that consists of only a few commands, e.g. small calculations) or as a “scratchpad” for testing pieces of code for a program you are writing. Interactive mode is available in the Python and IPython consoles in the bottom-right of the Spyder window. The differences between these are explained in the Spyder tutorial.

3.4 Writing Python programs

A program is a set of instructions written in a language that can be interpreted by the computer. Python programs normally have a *.py extension. The file extension is not important to Python, which will ignore the filename and just look at the contents. However, the extension is still quite important - most text editors used for programming will use the file extension to recognise that the file contains Python code (and not Matlab code, or Java, or C++, or your first novel). This will determine how they present and/or interpret your code.

You can write code in the editor window in Spyder. Save the file with a sensible filename with a *.py extension. Then you can run the program:

- *From the IDE:* Click the large right-arrow button in the menu bar at the top of the Spyder environment. This will execute the current program file in whichever console is currently open in the bottom right of the screen.
- *From a terminal:* A saved program called “myprogram.py” can be executed from a terminal by entering (e.g.) `python myprogram.py` at the command line.

3.5 Exercises

1. Move to either of the consoles in Spyder. Then try entering the following commands, noting the console output when you do so:

(a) `print("hello world!")`

(b) `print hello world!`

(c) `x=2`

(d) `y=3`

(e) `x`

(f) `y`

(g) `x+y`

2. In the editor, open a new file and write the command `print("hello world (again)")` on the first line. Save the file.

3. Execute the program from the IDE by clicking the large right-arrow above the editor. Note the output in the console window.
4. Leave Spyder and open a terminal. Navigate to the folder where you saved your program. Run your saved program by entering `python myprogram.py` (replacing "myprogram.py" with the filename you used).

4 Getting help

The remainder of the worksheets will take you through some basic skills in Python programming. As you work through the worksheets, there are a number of sources of help.

4.1 Using the ELE forum

You can use the forum on the ELE page for ECMM455 to get help from course staff and participants. The lecturer and demonstrators will monitor this forum and try to respond quickly. Please feel free to respond to other people's queries if you know the answer - or even just part of the answer. Use of the forum will speed up response time (someone may already have asked your question) and enable staff to answer each question once, rather than repeatedly over email. Please make use of this facility - an active forum benefits everyone! Use it as follows:

- Log in to the ELE page for ECMM455, find the forum titled "Programming in Python".
- If no one has asked your question, create a new topic e.g. "Worksheet 1, Exercise 6.8".
- Include enough relevant information for others to understand, and ideally, replicate the error, e.g. "I can't find the bug in this code: `print hello world`". Cut-and-paste from your code and the error message if this helps. [Note that the mouse right-click paste option won't work on ELE, use keyboard CTRL-V instead.]
- If you can help someone else, please do so by replying to their post.

4.2 Other sources of help

Some other sources of help are as follows:

- *Online tutorials*: There are lots of excellent Python tutorials online - a Google search will reveal them.
- *On the web*: Specific searches on Python commands or processes also usually return useful answers. Also cut-and-paste of an error message directly into Google search often leads to a solution. The Python user community is large, helpful and often quite geeky, so a lot of their knowledge ends up on the web. (Anything from a site called StackOverflow is usually pretty good - it's written by programmers, for programmers, and someone has almost certainly had the same problem as you. Just make sure you follow their etiquette - look for a solution yourself first, then look for previous posts on the same topic, *before* you ask a new question.)
- *Online documentation*: A lot of documentation is available at the Python website: <https://www.python.org/>. It is fairly comprehensive if you know what you are looking for.
- *From interactive mode*: Typing the command `help` will lead to a large set of documentation, but it's not as user friendly as the web.

4.3 A good book

The recommended course text for learning Python is “Think Python” by Allen Downey. Make sure you read the 2nd edition, which is updated for Python3. The reference:

Downey, A.B. (2015) Think Python. 2nd Edition. O'Reilly Media.