NSC1002: Mathematics and Computing; Integrative Tools for Natural Sciences

Introduction to Computing (with Python)

Text Books:

Introduction to Computation and Programming Using Python: John V. Guttag

Python for Scientists: John M. Stewart

Prof Geoffrey Vallis (Term 1) + TAs

Prof Beth Wingate (Term2) + TAs

Goals of the Computing/Programming part of this module

- Prepare students to advance through the Natural Sciences program
- Help students understand why computing and programming is important
- Help students feel genuinely confident of their ability to write useful computer programs.
- Prepare students to compete for jobs
- Help students understand how to create ‘algorithms’ from ‘problems’

Course timetable details

Tuesday: 2.35 – 4.25 (Harrison 207/208)
Friday: 12:35 – 2.25 (Harrison 108)

Five Quizzes:

Tentatively (dates and format to be confirmed)

24nd Nov, 1st Dec, 8th Dec + 2 more in spring

Summative (10% of total mark)

15 minutes, at the start of each class (probably online)

Be there! No excuses.

WHO ARE WE?

- Geoff Vallis (me)

- Laura Currie
- Paul Bowen
- Rebecca Millington
- Chris Fenton
- Kathie Hinton
- Surabhi Desai

- Next term: Prof Beth Wingate

Resources

1. We are trying to learn programming in this module, and Python is the language we choose to use. Once you have learned one language , learning other languages is easier.

2. Why Python?
   i) It is free.
   ii) It is easy to learn and use.
   iii) It embodies good coding practice.
   Iv) It is commonly used in the outside world, so a very employable skill.

3. We will be use “Python 3”, and we recommend Anaconda Python 3, with Spyder.

4. Books: “Python for Scientists, by John M. Stewart.
   “Introduction to Computation and Programming Using Python,” by John Guttag

5. Online documentation and videos:
   code academy, stackoverflow.com, tutorials point, etc. Use the web as a resource!

What should you expect from the Computing part of the module?

1. Write, (compile), test, and debug a computer program.
2. Document software to accepted standards.
3. Use a high-level programming language for basic numerical analysis, simulation and data analysis and visualization.
4. Systematically break down a problem into its components.
5. Understand and choose appropriate programming techniques

UNIVERSITY OF EXETER

Topics we will study over the course of this term

- Python as a community computer programming language
- Data types used in Python
- Plotting and graphics in Python
- Conditionals (if statements), while and for loops.
- Functions in computer programming
- Pseudo-code and debugging
- Numerical integration, numerical differentiation, matrix-vector operations and systems of equations
- Using community Python packages

Why is computing important?

Because writing, reading, and executing computer programs is ***essential*** for

- conducting science,
- analysing experiments,
- testing theory,
- making visual examples of scientific data
- writing reports
- creating presentations
- working in groups
- etc

Computing programming is probably the most employable skill you will learn at University.

How do I learn to program?

- Problem solving with computing is a skill that evolves with time.

1. It is more like learning to play a piano (with practice you can play more and more complex ‘pieces’) than studying for A-levels.

2. It is a ‘practice’ where your intuition grows and your skills sharpen the more you use it.

3. Program on your own time, ask your friends, use the Web!

On to Workshop 1 -- What is a computer language?

- A programming Language takes a human language and transforms it to a set of instructions a computer can understand (speaks in 1s and 0s).

1. Low-level (nearer machines) versus High Level (nearer humans)
2. Interpreted versus Compiled

Interpreted (Python, Matlab, etc):

Executes the code as it reads it.

Source Code > Checker > Interpreter > Output

Compiled (C, C++, Fortran), etc:

Looks at the entire code first and compiles it into machine code

Source Code > Compiler > Object Code > Interpreter > Output

Python is an *interpreted* language. It is also a *high-level* language.

Why python?

- About the origin of the name Python, Van Rossum wrote in 1996:

  "I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of *Monty Python's Flying Circus*)."

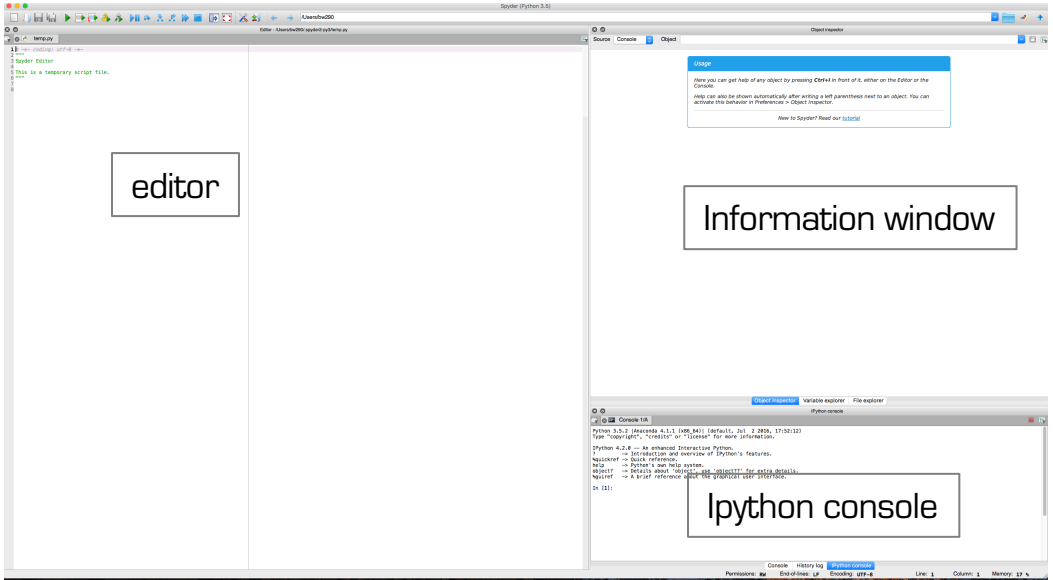
- Advantages
  - Widely Used (Google, Amazon, Science…), Freely Available, Community supported, Runs on most operating systems, interpreted, convenient and clever interfaces, easy to learn

There is no expectation that you will master these concepts today so

Relax!

Finding and using Spyder

- Log on to your computer.
- Click on the lower left search icon and type ‘spyder’ into the window.
- It will bring up a window that looks like this:



Using the ipython window (lower right corner)

- The text that appears:

```
In [1]:
```

- Type:

```
In [1]: 3 + 2 (press the rtn)
Out[1]: 5
```

- Type:

```
In [2]: 3. + 2.
Out[2]: 5.0
```

- Notice the difference – one has a decimal, the other does not. This is because the first one has a data type of `int` the other data type is a `float.`

Using the ‘type’ function

- Type into the ipython shell

```
In[ ]: type(2.)
Out[ ]: float

In[ ]: type(2)
Out[ ]: int
```

Using the editor

- The python editor is in the left panel of spyder
- The green area is for comments “”” “””
- Change the green area to include the following

```
Your Name
Program: My first program
```

- Then type in to the black area:

```
print(‘Hello World’)
```

- Click on the green arrow at the top.
- The output appears in the ipython console in the lower right.

Getting ‘help’
after a command type the question mark

```
>> print?

Docstring:
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file:  a file-like object (stream); defaults to the current sys.stdout.
sep:   string inserted between values, default a space.
end:   string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
Type:      builtin_function_or_method
```

Variables

- ‘Variables’ are created when they have something assigned to them. Add this to your file Test.py and run the module as before.

```
a = 1
print (“a”)
```

- Try this

```
a  = 3.
b =  -2.
print (“a + b = “,a+b)
```

- Can be an combination of letters and numbers + _

  Best to choose variable names with a meaning

Objects

- Objects are the core element that python programs manipulate. Every object has a type.
- Try typing the following into the interpreter

```
type (3 + 2)
type (3.0 + 2.0)
type ('this is a string')
type (True)
```

What is a data type?

- a particular kind of data item is defined by the values it can take
- In python the most common data types we use are `int`, `str`, `float`, `string`, and `bool`
- `Bool` is short for Boolean. A boolean data type means it is a binary variable that can have one of two possible values, 0 (false) or 1 (true).
- Reminder: In python you can use type() to check.
- When you typed in `print('Hello World')` the part 'Hello World' is of type `string`.

You should now know

- Know what to expect from this part of the module.
- Understand the importance of learning to program.
- Understand what Python is.
- Know how to start Spyder on the GP108 laptops
- Know where to find instructions for installing python on your own laptop.
- Prepare to experiment with Python programming using Spyder

Next time

- More about Spyder
- More data types
- Introduce lists, strings, and tuples and plotting
- Understand conditionals in programs (if statements)