Workshop 1 Introduction to Python

Prof. Geoffrey Vallis & Prof. Beth Wingate

3 November 2020

By the end of the session you will (hopefully!):

- Know what a program is
- Know what a programming language is
- Know what Python is
- Know what Sypder is, and how to use it to write simple computer programs
- Know where to find information regarding installing Python on your own machine
- Understand what a data type is

If you have any questions during this session please do not hesitate to ask one of the TAs (demonstrators). You should ensure you have completed all exercises before the next workshop.

What is a computer program?

FYI: program == programme (we use both spellings; usually just program!)

- A set of instructions that tells the computer what to do
- Everything a computer does is done by writing and running computer programs. This includes all applications such as Microsoft Word, Computer Games, Web Browsers such as Safari and Firefox.
- Machine learning and artificial intelligence are based on computer programs.
- Data analysis relies on computer programs.
- Video games are computer programs.

What is a programming language?

- A computer programming language is a language used by humans to write a list of instructions (a program) to tell a computer what to do.
- All programs like Safari and Firefox are written by humans using a computer program.

- Examples of computer programming languages are: Python, C, C++, Fortran, Matlab, Mathematica, and many others.

What is Python?

- Python is a particular programming language.
- In this part of the module we will use Python to create lists of instructions to tell the computer what to do. This is called a Python program.
- Python is installed by downloading it from the internet as a package. The precise way depends on whether you have a Windows machine or a Mac. We will use a particular distribution called Anaconda Python, because it is easy to install and comes with an environment called *Spyder* and a nice Python editor.

What is Spyder?

We use Spyder for convenience. You don't have to use it if you prefer something else.

- Spyder stands for "Scientific PYthon Development EnviRonment:" and is an application, like any other application, and comes with the Python package that you download and install.
- Start it just like you would start any other application, by clicking on icon or file menus. You can find it by typing "spyder" in the search window available on the lower left corner of the computer desktop.
- Spyder is an environment, meaning it is a collection of tools (applications) that can be used together. The Spyder application comes with the following tools:

  1. Spyder has a special **text editor** that is friendly to writing Python programs. It uses syntax highlighting, autocompletion, and smart indentation (left).
  2. Spyder has a special **iPython shell** that using highlighting (lower right).
  3. Spyder includes a **help console** (upper right).

Using the iPython shell:

1. Type a command into the lower right box of Spyder and *press* the return key:

```
In [1]: print('Hello World') <PRESS return/enter>
Hello World
```

2. The text that appears below the print statement and is produced when the Python interpreter evaluates the Python code entered at the prompt.

3. Type in the following :

```
In [2]: 3 + 2 <PRESS return/enter>
Out[2]: 5
In [3]: 3. + 2.<PRESS return/enter>
Out[3]: 5.0
```

Type in the text at the prompt In [2]:, press return, and Python will have run the code through its interpreter to produce the text labeled as Out [2].

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.

4. Type in the following :

```
In [*]: 3 // 2 <PRESS return/enter>
Out[*]: 1
In [*]: 3 / 2 <PRESS return/enter>
Out[*]: 1.5
```

From now on we use a $*$ notation to imply the iPython console. The console will place a number in that spot.

Note the double forward stroke, this means *integer* division, while a single stroke is for *floating point* division which is what we usually want.

5. Try the following:

```
In [*]: type(3.0) <PRESS return/enter>
Out[*]: float
In [*]: type(3) <PRESS return/enter>
Out[*]: int
In [*]: type('3') <PRESS return/enter>
Out[*]: str
```

The command "`type(3)`" is a function called "`type`" that tells the Python command interpreter to evaluate 3 and print its `type` out at the terminal. This concept is important so that Python knows what types of operations can be performed. These three data types are discussed more below in item 8 on page 4.

6. From now on pressing the `<return/enter>` will be implied from the context, but not explicitly stated. Type all the following commands and note Python's response:

```
In [*]: 3 + 7
In [*]: print(3 + 7)
In [*]: print('3+7')
In [*]: 3 + 5 * 8^2
In [*]: 3 + 5 * 8**2
In [*]: A = 2 * 2
In [*]: print(A)
In [*]: print(A + 2)
```

```
In [*]: B = A**2 - 3
In [*]: print("B = ", B)
In [*]: print(3, 'Hello')
In [*]: Greeting = 'Hello'
In [*]: Name = 'Alice'
In [*]: print(Greeting, Name)
In [*]: print(Greeting + Name)
In [*]: print(Greeting + '  ' + Name)
```

Any one of the above lines is a (rather simple) Python program because it is an instruction that tells the computer what to do. More complicated programs may be hundreds or thousands of lines long.

Feel free to enter other commands to see what is returned on screen. If you don't know, use a **search engine**. One primary advantage of using Python as a computing language is its international users community which includes tutorials, online documentation, and shared software.

7. Now type in the following command:

```
In [*]: print('Bob)
```

What happens? An error message will appear due to the missing '. Type instead,

```
In [*]: print('Bob')
```

This is known as a `syntax error` and occurs when the arrangement of the symbols in the program is incorrect (i.e., something occurs that Python did not expect or Python is missing something that it expected). Each error message will give you an indication of what is wrong (e.g. `EOL` refers to `End of Line`). Again, do not be afraid to use a **search engine** if you do not understand what the error is referring to.

8. Variables

Variables are created when they have something assigned to them. They can be used to save or store data to be manipulated later. Try the following:

Create the following variables:

```
In [*]: message = 'my pet snail is'
In [*]: age = 2
In [*]: count = 10
```

Identify what the following statements would do when entered into the prompt and then check to see if you were correct by entering them into the Python Shell.

```
In [*]: print(message + age)
In [*]: count * 2
In [*]: message
In [*]: name = 'Harold'
In [*]: joined = name + ',' + message)
In [*]: print(joined + age)
In [*]: count * 2 + 8
In [*]: age - 1
In [*]: count / 2
In [*]: count / (age * 4)
```

There are three major data types: integer, float, and string. Try to identify which data types of the above. A useful "function" in Python is called "`type()`". It will tell you the data "type" object or variable. You can change an int to a string by using the `str()` function.

```
In [*]: print(type(message))}
In [*]: print(type(age))}
In [*]: print(message + str(age))}
In [*]: print(joined + str(age))}
```

In Python we use operations on data types, such as $* + -/$. These work for strings, too. For example you can use + to add two strings. This is called "operator overloading". You can convert one type of data to another. For example trying converting an integer to a string using str(2). Try the following:

```
In [*]: type(3)
In [*]: type(3.)
In [*]: type('Harold')
```

Using Spyder's editor:

The editor is located in the left panel of the Spyder environment.

1. Navigate your mouse to this window. The python editor uses highlighting and font colors to aid in the development of programs. For example, the text highlighted at the top of the file is for a *docstring*, which holds text comments about the program. The docstring is located between TWO triple quotes: """. It is essential that you document your programs. To try the editor, type the following at the top of the editor:

   ```
   Your Name
   Assignment:  Program to try the Spyder Editor
   ```

2. Below the docstring, type in::
   `print('Hello World')`

Navigate your cursor to the upper bar and click on the green arrow, which will run the program through the python interpreter. Spyder will ask you to save your file. Save your file to the file name `Test.py`.

3. Try editing your program to output a different message. Run the newly edited program (click on the green arrow). You may want to save it in a new file.

   Write a new program (navigate your mouse to the `File` button, and selected `New` from the menu) . Enter the following text:

   ```
   message = 'my pet frog is'
   age = 42
   print(message + ' ' + age)
   ```

   Save the program, only this time create a new directory called `ProgrammingAssignments` before you save it. Did you get the output you expected?

   Introduce a syntax error (for example, remove one of the quote marks) into your program and see what happens when you run it. Make sure you understand the exception that has been thrown. To find out you can **debug** your program by putting in a print statement to print the `type` of each variable. Add the following to your program:

   ```
   message = 'my pet frog is'
   age = 42
   print('The data type of message is', type(message))
   print('The date stype of age is', type(age))
   print(message + ' ' + age)
   ```

   Click the green arrow. Remember, Python can only perform operations on data of the same type. You can convert `age` to a `sting` by using the function `str(age)` *before* you try to add them together, just as you did in exercise 9 that begins on page 4.

4. 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)
   ```

   Run your module by navigating your cursor to the top of the window and click on the green arrow.
   What happens?
   Next, type this into your program:

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

Write down the answer. Is it what you expected?

Variables can be an combination of letters and numbers. For long programs it is best to choose variable names with a meaning.

5. Exploring data types:

   A data type is a particular kind of data item, as defined by the values it can take. In Python the most common data types we use are int, 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.

   - The expression 3 + 2 is an object 5 of type int.
   - The expression 3.0 + 2.0 is an object 5.0 of type float.
   - In your program type this:
     ```
     print(3 != 2)
     print(3 == 3)
     ```
     Notice there are no quote marks!
   - Operators are: + - / * **
   - Order matters: Try evaluating 4*4**1-3 in your program and running it. Does it give what you expect?
   - Remember PEMDAS (Parenthesis, Exponentiation, Multiplication and Division, Addition and Subtraction)

Exercises

These exercises should be completed before the next workshop.

1. Explain the difference between `x = 2` and `x = '2'`.

2. Explain what the following statements would achieve before checking your answer using the Python shell:

```
In [*] x = 2
In [*] y = '2'
In [*] print(3 * x)
In [*] print(3 * y)
In [*] print(x * y)
In [*] a = b + c
```

3. Write a program using Spyder that defines a = 2.0 and b = 3.0 and prints out the value of a + b. Save the program with a new name, and execute it (for example, using the green arrow).

4. Write a program using Spyder `circle.py` that will calculate and print out the circumference of a circle for 5 different radii: r=1.0, 20.0, 65.0, 70.0, 100.0. In case you have forgotten, the formula for the circumference of a circle is, $c = 2\pi r$.

5. Write a program `fahrenheit.py` that converts temperatures from Celsius to Fahrenheit. In case you have forgotten, degrees Fahrenheit are obtained by multiplying degrees Celsius by 9/5 and adding 32. Your program should start with a variable that is initialised to the temperature in Celsius; then set another variable equal to the temperature in Fahrenheit and print the result (with an explanatory message).

6. What is a computer program?

7. What is a programming language? Give an example.

8. What is Spyder?

Installing Python and Editors

You can install Python by navigating to:
`https://www.anaconda.com/distribution/` And follow the instructions.

You don't have to use Spyder, or even Anaconda Python (there are other companies that distribute Python), but that is the one that we shall use in the class.

What you have learned today

By the end of the session:

- You know what a program is
- You Know what a programming language is
- You know what Python is
- You know what a Python shell is and how to use it
- You know know what Spyder is and how to write a program, save it, and run it.
- You have written simple Python programs using both the shell and the editor in Spyder;

- Know where to find information regarding installing Python on your own machine.

If you have any questions during this session please do not hesitate to ask one of the demonstrators. You should ensure you have completed all exercises before the next workshop.

You learn by doing! This is especially true of computer programming!