Python Programming

Joe Carthy

UCD School of Computer Science
University College Dublin
Ireland

Lesson 1
Programming, Algorithms and Output

- A computer program is written to solve a particular problem
- It is the programmer who solves the problem, not the computer
- The programmer gives the computer precise instructions on what to do: this is the program
- The program is a description to the computer of what it has to do and how to do it
- The program is the solution to the problem: when it is executed, we will get the required result
- The set of steps required to solve a problem is called the algorithm
- An algorithm written in a particular programming language is called a computer program

- What is an Algorithm?

- A set of instructions that, when executed, will solve a particular problem.

- Word comes from name of a Persian mathematician, Al-Khwārizmī who wrote *On the Calculation with Hindu Numerals* (circa 825 AD)

- Translated into Latin as *Algoritmi de numero Indorum* ("Al-Khwārizmī on the Hindu Art of Reckoning")

- A common real-world example (or approximation) of an algorithm is a cooking recipe!
- Recipe for Tea Brack (see http://odlums.ie/recipes/tea-brack)
- Ingredients
  - 225g Self-Raising Flour
  - 375g packet of Fruit Mix
  - 300ml cold Tea
  - 125g Caster Sugar
  - 1 Egg (beaten)
  - Good pinch Mixed Spice

- Method

1. Place fruit and tea in bowl and leave to soak overnight.
2. Add sugar, egg, flour and mixed spice and mix well.
3. Transfer to a greased and base-lined 900g loaf tin or a 20cm round cake tin.
4. Bake in a pre-heated oven (170°C/Gas Mark 3) for approximately one hour or until risen and firm to the touch.
5. Cool on a wire tray.

- A recipe is not really an algorithm, because. . .
- It is Imprecise
  - Lots of detail left out
  - How do you beat an egg?
  - What kind of tea?
  - What is a “good pinch”
  - Which shelf in the oven?
  - What is “overnight”?
  - “Approximately” one hour? “Risen”? “Firm to the touch”?
- It is Ambiguous
  - “Add sugar, egg, flour and mixed spice. . . ” Add to what?
  - Fan-assisted oven?
- Take it out of the oven! What does this mean?

An algorithm is a finite set of basic instructions, which, when executed, solve a problem.

- An algorithm should be precise
- An algorithm should be unambiguous
- An algorithm (normally) takes a defined set of inputs
- An algorithm (normally) produces a defined set of outputs
- An algorithm should terminate after a finite length of time
- An algorithm should guarantee to produce a correct result

When we have developed an algorithm, we need to express it in Python to run it on a computer.

We need to learn how to do basic operations in Python before we can develop significant programs.

We begin by learning how to display information on a screen, in Python.

We call this **output**.

A program to display a message on the screen:

```
print (  'Hello, world .' )
```

This causes Python to display the following output on the screen:

```
Hello, world.
```

The text in quotes is called **string** i.e.

```
Hello, world.
```

A string is made up of individual items called **characters**

Each letter is a character, comma is a character and dot is a character

To run (execute) this program, on my system:

The program is stored in a file which I call *print1.py*. You may use any name you wish. For example this file contains one line:

```
print(“I am Beth. This is my first program“)
```

To execute the program I use the command *python3* which runs the program and the output of the program is displayed on the screen:

```
% python3 print1.py
I am Beth. This is my first program
```

A single `print` command can output a number of strings. It displays them, separated by a space, in the order in which they appear

For example, `print` in the program below, displays 2 strings

```
print ('Good morning,’, 'Vietnam !’ )
```

It produces the following output on the screen:

```
Good morning, Vietnam!
```

Comments

- Any text following the # character is ignored by Python
- This text is called a comment
- Comments are used to make programs easier to understand

```
# My first program
print('Hello ,   world .’ )
```

This programs displays the following output on the screen:

```
Hello, world.
```

```
#  My  second   program
print('Good morning!')
print('Vietnam!` )
print('Good  morning,',  'Vietnam!')
```

This causes Python to display the following output on the screen:

```
Good morning!
Vietnam!
Good morning, Vietnam!
```

- Each line of your program will have a **statement** or a comment.

- `print` is an example of a statement. It is a command to Python.

- You must be precise when writing statements.

- It is an error to omit a bracket or quote.

- It also an error if you misspell a command like `print` - say you spell it as `prnt`.

- Such errors are called **syntax errors**.

- Beginners make many syntax errors – it’s nothing to worry about but you must fix them before you run your programs.

- If you make such an error:

  check your spelling of each command

  check that for **every opening** bracket or quotation mark that you have a corresponding **closing** bracket or quotation mark.

Running your Python programs

- Most programmers use an integrated development environment (IDE) to enter and run their programs
- The IDE provides:
  - Allows you enter and store your program in a file
  - Allows you run your program
  - Helps you find errors in your program (code - we sometime say ***code*** for program and use the word ***coding*** for programming)

  - Two commons IDEs are **IDLE** and **Thonny**.

- There are several IDEs that you can download from the web. Each IDE provides the user with menus such as
- The file menu
  - Create a new editing window
  - Open a file containing an existing Python program
  - Save the contents of the current editing window into a file (with a **`.py`** file extension)
- The edit menu
  - Standard text-editing commands
  - Copy, paste, find, . . .
  - Commands for editing Python code
  - Indent a region of code

Time to practice !

- Copy all of the examples from the slides above and get them to run in your Python environment.

- Then complete the exercises from the Handbook and get them to run.

- Finally carry out the assignments from the Handbook and get them to run.