1811/2807/7001ICT
Programming Principles

School of Information and Communication Technology
Griffith University

Trimester 1, 2024

2 About Python 3

This section is about aspects of the Python 3 programming language and where to get its tools and documentation.

2.1 Language characteristics

Python 3 is a *programming language*.

> *Definition*: A programming language is an artificial language for describing actions to be performed by a computer, that is readable by *both* human programmers and computers.

Other example programming languages: assembly language; Fortran; Fortran; COBOL; BASIC; Pascal; C; C++; Java; C#; JavaScript; Haskell; Swift.

Not a programming language: HyperText Markup Language (HTML) – why not?

Python 3 is a *high-level programming language*.

> *Definition*: High-level programming languages are languages that are structured for more human readability than lower level languages, such as assembly language.

High-level languages start with the third generation of programming languages.

The first high-level, third-generation programming language was Fortran (formula translation).

Python 3 is a is an *interpreted programming language*. The `python3` program is an interpreter.

> *Definitions*: An interpreter is an alternative to a compiler.
>
> An interpreter reads a program and executes it immediately.
>
> A compiler reads the program and writes it out again in a different language, without executing it. The translated program is run later.

A program that is compiled will run faster than a program that is being interpreted.

Languages that are compiled: assembly language (the simple compiler is called an assembler); Fortran; COBOL; Pascal; C; C++; Java; C#; Haskell; Swift.

Other languages that are interpreted: BASIC; JavaScript.

Python 3 is a *hybrid programming language*, mixing the attributes of the imperative programming paradigm, object-oriented programming, and functional programming.

Older and lower-level programming languages are usually purely imperative languages, such as assembly language, Fortran, BASIC, Pascal, C. An imperative is a command. Imperative programs are mostly sequences of commands.

Most object oriented languages are extensions of and hybrids with imperative languages, such as C++, Java, C#, JavaScript. These languages add classes and objects to make larger programs easier to manage with more reusable components.

Pure functional programming languages do not consist of sequences of commands. Data is transformed from inputs to outputs according to relationships defined by functions. Haskell is an example of a modern purely functional programming language.

Functional programming can implemented in any imperative language, some mostly imperative languages, have some extra support for programming in the functional style, particularly Python and Swift, which also are object-oriented.

2.2 Python 3 versus Python 2

For this course we will only use version 3 of the Python language.

There are enough changes between them that Python 2 programs can not be interpreted by the Python 3 interpreter.

Python 2 code will not be accepted for marking in this course.

From here, just *Python* always refers to Python 3.

2.3 Language resources

The primary resource for Python programmers is the Python website: `https://www.python.org/`.

From here you can download software and access the documentation, particularly that of the standard library.

2.4 Installing Python

Download the Python software installer from https://www.python.org/ and run it.

It installs slightly differently on different operating systems.

MacOS: MacOS already has Python 2 installed by default.

The interpreter command for Python 2 is `python`.

The interpreter command for Python 3 therefore has to be different and is `python3`.

Windows: Python is not preinstalled on Windows, so the Python 3 interpreter is just called `python`.

The Windows install wizard for Python 3 has two pages of options.

Most of them should be left with their default values.

On the second page check “Add Python to environment variables”.

This ensures that you can type `python` on the command line.

If you forget to do this, run the installer again to modify the settings.

2.5 Running Python

There are several ways to write and run Python code.

- The Python interpreter (`python` or `python3`) is an interactive command line tool, a *read-eval-print loop ()*.

  > *Definition*: A read-eval-print loop is an interactive program that reads lines of code typed by the user, evaluates or executes them, prints the results, and loops back to reading.

- The Python interpreter can run a *script* of saved Python code.

  > *Definition*: An script is a (usually) small program expressed in an interpreted programming language.

- You can use an *integrated development environment ()*, such as:
  - `IDLE` (comes with Python 3); or
  - PyCharm.

    Definition: A IDE is a software application that presents a program editor, compiler, debugger in a single user interface.

It is very important that you become capable of using Python in *all* of these ways.

2.5.1 REPL

Example command line session:

```
$ python3
Python 3.6.2 (v3.6.2:5fd33b5926, Jul 16 2017, 20:11:06)
Type "help", "copyright", "credits" or "license" for
more information.
>>> 3 + 4
7
>>> quit()
$
```

(Appendix A of the lecture notes explains how to read this example.)

REPL mode is best for quick play and testing of ideas.

2.5.2 Scripts

Python code is (usually) saved in text files with the extension `.py`.

Most of the example Python code shown in this section is supplied in scripts in a zip file you can download from Learning@Griffith.

Most of the examples that follow use code in scripts and run with the `python3` tool.

2.6 Things to do

In our first workshop we will show you how to use the Python REPL.

Then, in your own time, download and install Python 3 on your own machines.

Section summary and further reading

In this section:

- Python programming language;
- a high-level programming language;
- an interpreted programming language;
- an imperative/object-oriented/functional hybrid programming language;
- where to get the Python software and documentation; and
- the ways Python programs can be run.

For this section you should:

- explore the Python web site, https://www.python.org/.