DAY-1

Python Introduction

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

PYTHON SYNTAX

Python Indentation

- Indentation refers to the spaces at the beginning of a code line.
- Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.
- Python uses indentation to indicate a block of code.
- Python will give you an error if you skip the indentation.
- The number of spaces is up to you as a programmer, but it has to be at least one.

Python Variables

- In Python, variables are created when you assign a value to it.
- Variables are containers for storing data values.
- Unlike other programming languages, Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.
- Variables do not need to be declared with any particular type and can even change type after they have been set.
- String variables can be declared either by using single or double quotes.
- **Rules for Variable Names:**

  A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume). Rules for Python variables:
  - A variable name must start with a letter or the underscore character
  - A variable name cannot start with a number
  - A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
  - Variable names are case-sensitive (age, Age and AGE are three different variables)
- **Assign Value to Multiple Variables**
  - Python allows you to assign values to multiple variables in one line
  - And you can assign the *same* value to multiple variables in one line
- **Output Variables**
  - The Python `print` statement is often used to output variables.
  - To combine both text and a variable, Python uses the + character
  - You can also use the + character to add a variable to another variable
  - For numbers, the + character works as a mathematical operator
  - If you try to combine a string and a number, Python will give you an error
- **Global Variables**
  - Variables that are created outside of a function (as in all of the examples above) are known as global variables.

        - Global variables can be used by everyone, both inside of functions and outside.
        - If you create a variable with the same name inside a function, this variable will be local, and can only be used inside the function. The global variable with the same name will remain as it was, global and with the original value.
- **The global Keyword**
    - Normally, when you create a variable inside a function, that variable is local, and can only be used inside that function.
    - To create a global variable inside a function, you can use the `global` keyword.
    - Also, use the `global` keyword if you want to change a global variable inside a function.

Python Comments

- Comments can be used to explain Python code.
- Comments can be used to make the code more readable.
- Comments can be used to prevent execution when testing code.
- Python does not really have a syntax for multi line comments.
- To add a multiline comment you could insert a `#` for each line:

Multiline Comments

- Python will ignore string literals that are not assigned to a variable, you can add a multiline string (triple quotes) in your code, and place your comment inside it.

Python Data Types

- **Built-in Data Types**
    - In programming, data type is an important concept.
    - Variables can store data of different types, and different types can do different things.
    - Python has the following data types built-in by default, in these categories:

Text Type:	`str`
Numeric Types:	`int`, `float`, `complex`
Sequence Types:	`list`, `tuple`, `range`
Mapping Type:	`dict`
Set Types:	`set`, `frozenset`
Boolean Type:	`bool`
Binary Types:	`bytes`, `bytearray`, `memoryview`

- **Getting the Data Type:**
  - You can get the data type of any object by using the `type()` function
- **Setting the Data Type:**

  In Python, the data type is set when you assign a value to a variable.

Example	Data Type
x = "Hello World"	str
x = 20	int
x = 20.5	float
x = 1j	complex
x = ["apple", "banana", "cherry"]	list
x = ("apple", "banana", "cherry")	tuple
x = range(6)	range
x = {"name" : "John", "age" : 36}	dict
x = {"apple", "banana", "cherry"}	set
x = frozenset({"apple", "banana", "cherry"})	frozenset
x = True	bool
x = b"Hello"	bytes
x = bytearray(5)	bytearray
x = memoryview(bytes(5))	memoryview