# Introduction to Python

## *What is Python ?*

**Python** is a high-level general-purpose dynamic multi-paradigm programming language that emphasises on a clean and simple syntax that makes it largely orthogonal, yet programs written using this language are highly readable and maintainable.

## *Features of Python*

Now, let me elaborate on the features of this programming language that allows you to understand most of the jargons quoted in the above definition.

### High level programming language

Python programming cater to more general audience than computer science engineers. You do not need to understand computer architecture, memory model, CPU register-width, platform endian-ness and all those “computer engineering” centric jargons to write programs in Python. More over, the language does not overwhelm you with “pointers” like in C, or complex OO-jargons (abstract classes, interfaces, generics and so on) from languages like Java. This clearly means that python allows you to focus more on the problem domain while writing your program than the programming language and platform

### General purpose programming language

Python can be used to write programs that deal with scientific computing, data processing, network and process automation, web development, GUI development, game development and so on. It has all the features that a programmer needs to become productivity in developing applications for a specific domain. Though Python aims to be simple in syntax, it supports rich set of standard built-in and library functions that allow developing programs employing complex algorithms.

### Dynamic programming language

Python is designed to be a fully dynamic programming language. Python programs are essentially interpreted and this allows them to be more flexible at run-time. Developing applications that use complex design patterns and algorithms is easier in python when compared to many other static programming languages.

A dynamic programming language could imply two paradigms:

#### ***Dynamic typing***

A dynamic typing in a  programming language allows programmers to use variables and functions within their programs without declaration. This emphasises on programmer productivity. In Python, variables can be used directly after first defining them. Remember that definition is different than declaration - we will discuss more about that in the later chapter. One of the main advantages of dynamic typing is that the language decides the storage size and type of the variable, so that the programmer can focus more on functionality than language intricacies.

> Dynamic typing however expects programmers to be self-disciplined. Typos in variable names for instance, could lead to duplicate definitions and logical errors in programs that might become a bit difficult to detect without proper testing.

#### ***Dynamic programing***