Python Programming Language

Python is an easy to learn, powerful programming language.
It has efficient high-level data structures and a simple but effective approach to object-oriented programming.
Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

History

- Python laid its foundation in the late 1980s.
- The implementation of Python was started in the December 1989 by Guido Van Rossum at CWI in Netherland.
- In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.
- In 1994, Python 1.0 was released with new features like: lambda, map, filter, and reduce.
- Python 2.0 added new features like: list comprehensions, garbage collection system.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify fundamental flaw of the language.
- ABC programming language is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.
- Python is influenced by following programming languages:
    ABC language.
    Modula-3

Story behind the name Python

- Guido van Rossum, the creator of the Python language, named the language after the BBC show "Monty Python's Flying Circus".
- He doesn't particularly like snakes that kill animals for food by winding their long bodies around them and crushing them.

Python Releases

Python Version	Released Date
Python 1.0	January 1994
Python 1.5	December 31, 1997
Python 1.6	September 5, 2000
Python 2.0	October 16, 2000
Python 2.1	April 17, 2001
Python 2.2	December 21, 2001
Python 2.3	July 29, 2003
Python 2.4	November 30, 2004
Python 2.5	September 19, 2006
Python 2.6	October 1, 2008

Python Version	Released Date
Python 2.7	July 3, 2010
Python 3.0	December 3, 2008
Python 3.1	June 27, 2009
Python 3.2	February 20, 2011
Python 3.3	September 29, 2012
Python 3.4	March 16, 2014
Python 3.5	September 13, 2015
Python 3.6	December 23, 2016
Python 3.6	December 23, 2016
Python 3.7	July 8, 2019

Features of Python

Simple
- Python is a simple and minimalistic language.
- Reading a good Python program feels almost like reading English, although very strict English!
- This pseudo-code nature of Python is one of its greatest strengths.
- It allows us to concentrate on the solution to the problem rather than the language itself.

Easy to Learn
- As we will see, Python is extremely easy to get started with.
- Python has an extraordinarily simple syntax, as already mentioned.

Free and Open Source
- Python is an example of a FLOSS (Free/Libré and Open Source Software).
- In simple terms, we can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs.
- FLOSS is based on the concept of a community which shares knowledge.
- This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

High-level Language
- When we write programs in Python, we never need to bother about the low-level details such as managing the memory used by your program, etc.

Portable
- Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms.
- All our Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

- We can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!

Interpreted
- A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options.
- When we run the program, the loader copies the program from hard disk to memory and starts running it.
- Python, on the other hand, does not need compilation to binary.
- We just run the program directly from the source code.
- Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of our computer and then runs it.
- According to which python is compiled as well as interpreted.
- All this, actually, makes using Python much easier since we don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since we can just copy your Python program onto another computer and it just works!

Object Oriented
- Python supports procedure-oriented programming as well as object-oriented programming.
- In procedure-oriented languages, the program is built around functions which are nothing but reusable pieces of programs.
- In object-oriented languages, the program is built around objects which combine data and functionality ie (Characteristics and Behaviour).
- Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

Extensible
- If we need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, we can code that part of our program in C or C++ and then use it from your Python program.

Embeddable
- We can embed Python within our C/C++ programs to give scripting capabilities for our program's users.

Extensive Libraries
- The Python Standard Library is huge indeed.
- It can help us to do various things involving regular expressions,documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff.
- Remember, all this is always available wherever Python is installed.
- Besides the standard library, there are various other high-quality libraries which you can find at the Python Package Index.