Python

“Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.”

Python was conceived in the late 1980s and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to the ABC (programming language) capable of exception handling and interfacing with the Amoeba operating system. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL)Python was named for the BBC TV show Monty Python's Flying Circus.

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.

Easy to Learn

Python is extremely easy to get started with. Python has an extra ordinarily simple syntax

Free and Open Source

Python is an example of a FLOSS (Free/Libré and Open Source Software). You 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.

High-level Language

When you write programs in Python, you 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 your 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.

You 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

Python does not need compilation to binary. You 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 your computer and then runs it.

Object Oriented

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality.

Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java

Extensible

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

Embeddable

You can embed Python within your C/C++ programs to give scripting capabilities for your program's users

Extensive Libraries

The Python Standard Library is huge indeed. It can help you 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.

Applications of Python

1. **GUI-Based Desktop Applications:**

Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI).

2. Image Processing and Graphic Design Applications:

Python has been used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Lightwave and Maya, also use Python in variable proportions.

3. **Scientific and Computational Applications**:

The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modeling software, such as FreeCAD, and finite element method software, such as Abaqus, are coded in Python.

4. Games:

Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney's Toontown Online, Vega Strike etc.

5. **Web Frameworks and Web Applications:**

Python has been used to create a variety of web-frameworks including CherryPy, Django, TurboGears, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with database and interfacing with different internet protocols such as HTTP, SMTP, XML-RPC, FTP and POP. Plone, a content management system; ERP5, an open source ERP which is used in aerospace, apparel and banking; Odoo – a consolidated suite of business applications; and Google App engine are a few of the popular web applications based on Python.

6. **Enterprise and Business Applications:**

With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Reddit, which was originally written in Common Lips, was rewritten in Python in 2005. Python also contributed in a large part to functionality in YouTube.

7. **Operating Systems:**

Python is often an integral part of Linux distributions. For instance, Ubuntu's Ubiquity Installer, and Fedora's and Red Hat Enterprise Linux's Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

8. Language Development:

Python’s design and module architecture has influenced development of numerous languages. Boo language uses an object model, syntax and indentation, similar to Python. Further, syntax of languages like Apple’s Swift, CoffeeScript, Cobra, and OCaml all share similarity with Python.

Setting up the development environment

- Download the latest version of Python 3.6 from the official website.

  https://www.python.org

  If you want to be sure you are installing a fully up-to-date version, click the **Downloads** > **Windows** link from the home page of the Python.org web site.

- Add python installation location to path variable

  By design, Python installs to a directory with the version number embedded, e.g. Python version 3.6 will install at C:\Python36\, so that you can have multiple versions of Python on the same system without conflicts.

  It also does not automatically modify the PATH environment variable, so add the directories for your default Python version to the PATH.

Steps:

- Starting at My Computer go to the python installed directory In that folder you should see all the Python files.
- Copy that address
- Click on Start. Right Click on My Computer.

- Click on Properties. Click on Advanced System Settings or Advanced.
- Click on Environment Variables.
- Under System Variables search for the variable Path.
- Select Path by clicking on it. Click on Edit.
- Scroll all the way to the right of the field called Variable value using the right arrow.
- Add a semi-colon (;) to the end and paste the path (to the Python folder) that you previously copied. Click OK.

Writing Your First Python Program

Create a folder called PythonPrograms on your C:\ drive. You will be storing all your Python programs in this folder.

Go to Start and either type Run in the Start Search box at the bottom or click on Run.

Type in notepad in the field called Open.

In Notepad type in the following program exactly as written:

```
# File: Hello.py

print "Hello World!"
```

Go to File and click on Save as.

In the field Save in browse for the C: drive and then select the folder PythonPrograms.

For the field File name remove everything that is there and type in Hello.py.

In the field Save as type select All Files

Click on Save. You have just created your first Python program.

Running Your First Program

Go to Start and click on Run.

Type cmd in the Open field and click OK.

A dark window will appear. Type cd C:\ and hit the key Enter.

If you type dir you will get a listing of all folders in your C: drive. You should see the folder PythonPrograms that you created.

Type cd PythonPrograms and hit Enter. It should take you to the PythonPrograms folder.

Type dir and you should see the file Hello.py.

To run the program, type python Hello.py and hit Enter.

You should see the line Hello World!

Congratulations, you have run your first Python program.

Using IDLE on Windows

- Start IDLE
- Go to File menu and click on New Window
- Type your program in
- Go to File menu and click on Save. Type in filename.py This will save it as a plain text file, which can be opened in in any editor you choose (like Notepad or TextEdit).
- To run your program go to Run and click Run Module