

# 01-Python Basic

Introduction, Python Standard Library, REPL, Byte code compilation etc

Posted on January 29, 2023

Last updated on January 29, 2023

## 1.1 Introduction

Python is an open-source, platform independent, object-oriented, interpreted, high-level programming language with dynamic semantics.

It was created by **Guido van Rossum**

([https://en.wikipedia.org/wiki/Guido\\_van\\_Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum)) and first released in the year 1991.

Python's simple, easy to learn syntax emphasizes on code readability with the use of whitespaces and therefore reduces the cost of program maintenance.

Python supports modules and packages, which encourages program modularity and code reuse.

In January 1994, `map()`, `filter()`, `reduce()`, and the lambda operator were added to the language.

## 1.2 Python Name

The name Python was selected from the TV Show "The Complete Monty Python's Circus", which was broadcasted in BBC from 1969 to 1974.

Guido developed Python language by taking almost all programming features from different languages

1. Functional Programming Features from C
2. Object Oriented Programming Features from C++
3. Scripting Language Features from Perl and Shell Script
4. Modular Programming Features from Modula-3

Most of syntax in Python Derived from C and ABC languages

## 1.3 Where we can use Python

- System Programming
- Graphical User Interface
- Web Scrapping
- Web development
- Managing Database
- Fast Prototyping
- Numeric / Scientific Programming
- Game development
- Image Processing
- Robotics
- Automation
- Data science
- Data Visualization
- Data Mining

### NOTE:

- Internally Google and Youtube use Python coding
- NASA and New York Stock Exchange Applications developed by Python.
- Top Software companies like Google, Microsoft, IBM, Yahoo using Python.

## 1.4 Python Versions:

- Python 1.x introduced in Jan 1994
- Python 2.x introduced in October 2000
- Python 3.x introduced in December 2008

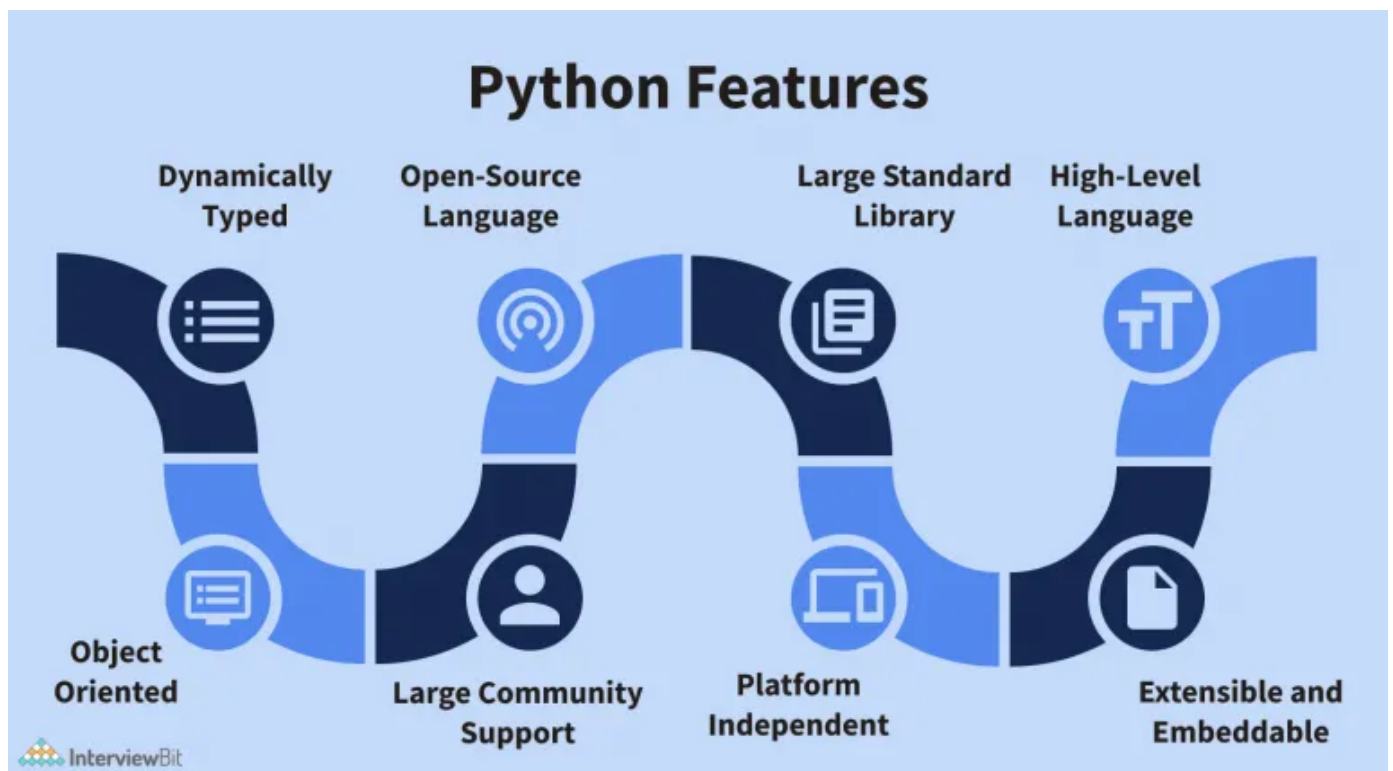
### NOTE:

- Python 2.7 was published on July 3, 2010 and planned as the last of the 2.x releases
- Python 3 won't provide backward compatibility to Python2
  - i.e there is no guarantee that Python2 programs will run in Python3.

## 1.5 Flavors of Python

- **CPython**
  - It is the standard flavor of Python. It can be used to work with C language Applications
- **Jython or JPython**
  - It is for Java Applications. It can run on JVM
- **IronPython**
  - It is for C#.Net platform
- **PyPy**
  - The main advantage of PyPy is performance will be improved because JIT compiler is available inside PVM.
- **RubyPython**
  - For Ruby Platforms
- **AnacondaPython**
  - It is specially designed for handling large volume of data processing

## 1.6 Features of Python



- **Simple and easy to learn** Python is a simple programming language. When we read Python program, we can feel like reading English statements.
- **Open Source And Free:** Python is developed under an OSI-approved open source license. Hence, it is completely free to use, even for commercial purposes

- **Large Standard Library:** Python has an extensive standard library available for anyone to use. There are libraries for image manipulation, databases, unit-testing, expressions and a lot of other functionalities.
- **Platform Independent and Portability:** We write a program in Python, it can run on a variety of platforms, for instance, Windows, Mac, Linux, etc. We do not have to write separate Python code for different platforms.
- **Extensible and Embeddable:** Python is an Embeddable language. We can write some Python code into C or C++ language and also we can compile that code in C/C++ language. Python is also extensible. It means that we can extend our Python code in various other languages like C++, etc. too.
- **Large Community Support:** With one of the biggest communities on StackOverflow and Meetup, Python has gained its popularity over the years. If we need any kind of help related to Python, the huge community is always there to answer our queries.
- **High Level Programming language:** A high-level language (HLL) is a programming language that enables a programmer to write programs that are more or less independent of a particular type of computer.

### **Limitations of Python**

1. Performance wise not up to the mark coz it is interpreted language.
2. Not using for mobile Applications

## **1.7 Why to use Python?**

Among numerous languages available in the market why should you choose python? This is the first question that arises in the mind of new users.

Following are the some of the reasons why people select python-

- **Quality of software:** Python was meant for readability. Its reusable and maintainable as compared to other languages. Its easier to understand. It supports all the modern features like OOPs and functional programming.
- **Productivity of Developers:** The same program which is written in other high-level languages like c++ or java can be written in one-third or one-fifth line of codes. That means debugging can be easy and it will be less prone to error which in turn increases the productivity of the developers.

- **Portability:** Mostly it's platform-independent . It can run on any platform or OS with minor or no change at all which makes it a highly portable language. Now you can use MircoPython to interact with hardware as well. It can be used on most of the edge devices.
- **Supporting Libraries:** Python already has a lot of inbuilt libraries that come with the standard python package which you download from its official site. With these libraries, you can build lots of basic applications or day to day automation tasks like copying data in bulk from one place to another. Apart from this, there's a huge list of third-party libraries like Numpy, Matplotlib, Scikit Learn, etc.
- **Fun to use:** Its simplicity and availability of lots of supporting libraries plus huge open source community support make development in python a breeze. That's why its widely preferred by hobbyists as well.

## 1.8 Zen of Python -

By importing this it displays the basic design philosophies of python

The Zen of Python, by Tim Peters

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

## 1.9 Python Library

- Python standard library consists of more than 200 core modules
  - The Python Standard Library is a collection of script modules accessible to a Python program
  - It simplify the programming process and removing the need to rewrite commonly used commands.
  - Eg:
    - Build-in data types(int, str,..)
    - File, OS operations(os, csv..) ..etc..
- Python has more than 137,000 libraries

## 1.10 Compiled language vs interpreted language

- If you have souce code it will run on any machine
- Dosen't matter weather source code is written in compiled/interpreted language

### Compiled Language

- Faster and more efficient to execute than interpreted languages.
- Errors are handled at compile time

Process of compiled language like C, Java

1. Write source code
2. Compiler turns source code into machine language that the CPU understands
3. Compiled machine language, you no longer need either the source code or the compiler
4. But different family of CPUs uses a different machine language, like Windows and Mac uses different compiler
  - So, the compiled code will only run on one family of machines

### Interpreted language

- Platform independence, dynamic typing and smaller program size
- Errors are handled at run time

Process of Interpreted language like PHP, Ruby, Python, and JavaScript

1. Write source code
2. Interpreter directly executes the statements, without compiling it to machine language.
3. So to run code in an interpreted language, you must have both the source code and the interpreter

# Python is Compiled or Interpreted language ?

- Python is interpreted language is half correct
- The source code is converted into bytecode(.pyc) that is then executed by the Python virtual machine.

## Byte code

- Byte code is the lower-level form of your program after Python compiles it.
- Python automatically compiles and stores byte code in files with a .pyc extension.
- Byte code files in a subdirectory named \_\_pycache\_\_

## Python compilation

1. Write source code
2. Python interpreter first compiles the source code(.py) into byte code(.pyc)
3. The Python interpreter then has a virtual machine that can run the byte code to machine language
4. The py or .pyc file can be used on any machine, py/pyc and interpreter is required

## 1.11 Identifiers

- A name in Python program is called identifier.
- It can be class name or function name or module name or variable name

### Rules to define identifiers in Python:

- The only allowed characters in Python are
  - alphabet symbols (either lower case or upper case)
  - digits (0 to 9)
  - underscore symbol(\_)
- Identifier should not starts with digit
- Identifiers are case sensitive. Of course Python language is case sensitive language.
- Should not use reserved words as identifiers

### General rule

- **Avoid** using names that are too general or too wordy.
  - Strike a good balance between the two.
- **Bad:**
  - data\_structure, my\_list, info\_map,  
dictionary\_for\_the\_purpose\_of\_storing\_data\_representing\_word\_definitions
- **Good:** user\_profile, menu\_options, word\_definitions

Don't be a jackass and name things "i", "j", or "k"

When using CamelCase names . capitalize all letters of an abbreviation (e.g. HTTPServer)

## Naming Conventions

- Classes

- *Model, MyClass*

- Variables (global and local)

- *var, my\_variable*

- Methods and Functions

- *function, my\_function*
  - *class\_method, method*

- Method Arguments

- *Instance methods should have their first argument named 'self'.*
  - *Class methods should have their first argument named 'cls'*

- Constants

- *CONSTANT, MY\_CONSTANT, MY\_LONG\_CONSTANT*

## 1.12 Reserved Words

In Python some words are reserved to represent some meaning or functionality. Such type of words are called Reserved words.

There are 33 reserved words available in Python.

```
import keyword
keyword.kwlist

['False', 'None', 'True', 'and', 'as', 'assert', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else',
'except', 'finally', 'for', 'from', 'global', 'if',
'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or',
'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

### Note:

1. All Reserved words in Python contain only alphabet symbols.
2. Except the 3 reserved words(True,False,None), all contain only lower case alphabet symbols.



## 1.13 Misc

### REPL

- Python Terminal
- Read , Evaluate , Print , Loop

### Python name scopes

- Names are looked up in four nested scopes
- LEGB rule : Local, Enclosing, Global, and Built-ins
- **Local**
  - Inside current fxn
- **Enclosing**
  - Any and all enclosing fxn
- **Global**
  - Top-level of module
- **Build-in**
  - Provided by the builtins module

### NOTE

- In Python, pretty much everything is an object , whether it is a number, a function or even modules.

.

.

.

.

.

.

.

.

.

.