**Python:**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Everything in python is object oriented.

**Origin:**

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and UNIX shell and other scripting languages.

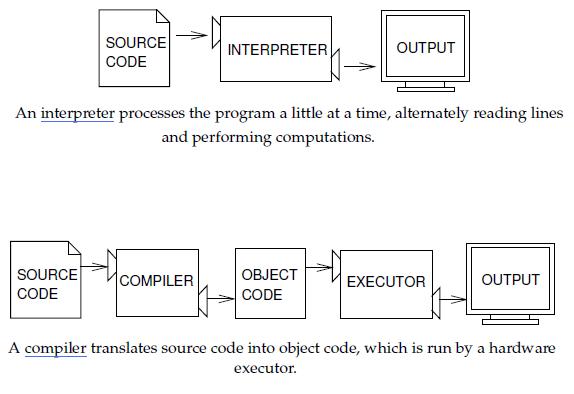
**Features of Python:**

* **Python is interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.
* **Easy-to-learn:**Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read:**Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain:**Python's source code is fairly easy-to-maintain.
* **A broad standard library:**Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode**: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable:**Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable:**You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases:**Python provides interfaces to all major commercial databases.
* **GUI Programming:**Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable:**Python provides a better structure and support for large programs than shell scripting.
* **Interpreted & OOP’s:** Python is an interpreter, object-oriented, high-level programming language
* **Byte code:** It can be used as a **scripting language** or can be **compiled to byte-code** for building large applications.
* **Dynamic**: It provides very high-level dynamic data types and supports dynamic type checking.
* **Automatic Garbage Collection:** IT supports automatic garbage collection.

**Compiler vs Interpreter:**

A [compiler](http://en.wikipedia.org/wiki/Compiler)takes entire program and converts it into object code which is typically stored in a file. The object code is also refereed as binary code and can be directly executed by the machine after linking. Examples of compiled programming languages are [C](http://www.geeksforgeeks.org/c/) and [C++](http://www.geeksforgeeks.org/c-plus-plus/).

An [Interpreter](http://en.wikipedia.org/wiki/Interpreter_%28computing%29)directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code. Examples of interpreted languages are Perl, Python and Matlab.



|  |  |  |
| --- | --- | --- |
| **No** | **Compiler** | **Interpreter** |
| 1 | Compiler Takes Entire program as input | Interpreter Takes Single instruction as input. |
| 2 | Intermediate Object Code / byte code is Generated | No Intermediate Object Code is Generated |
| 3 | This object code is used by cpu for execution. | Source code itself line by line is directly interpreted to binary for execution. |
| 4 | Source code (**Hello.java**) is un known because (**Hello.class**) byte code is sufficient for executing. | Source code is open because it is directly interpreted on machine. |
| 5 | Memory Requirement : More(Since Object Code is Generated) | Memory Requirement is Less |
| 6 | Errors are displayed after entire program is checked | Errors are displayed for every instruction interpreted (if any) |
| 7 | Example : Java Compiler | Example : Unix shell, Python Interpreter |

Note: Python source code (.py) can also be compiled to byte code (.pyc) in large scale applications by using external libraries as shown below hence it is called hybrid language. In most of the use cases source code is executed in interpreted mode.

import py\_compile

py\_compile.compile("file.py")

Applications build using Python:

1. Console Based Application
2. Audio or Video based Applications
3. 3D CAD Applications
4. Web Applications
5. Enterprise Applications
6. Applications for Images

Execution Modes in Python:

Interactive Mode: Execute commands directly from the python prompt. Typing python in Linux shell will take you to python prompt / Interactive Mode.

Script Mode: Write the code in file and save it as filename.py and execute as below.

Method 1: ./sample.py

Method 2: python sample.py

Variables:

Memory Management of objects in Python: