

PYTHON PROGRAMMING

http://docs.python.org/

The screenshot shows a web browser window displaying the Python v2.6.1 documentation. The browser's address bar shows the URL `http://docs.python.org/`. The page title is "Overview — Python v2.6.1 documentation". The page layout includes a left sidebar with navigation links and a main content area with a welcome message and a list of documentation sections.

Overview — Python v2.6.1 documentation

Python v2.6.1 documentation » modules | index

Download

Download these documents

Other resources

- FAQs
- Introductions
- Guido's Essays
- New-style Classes
- PEP Index
- Beginner's Guide
- Topic Guides
- Book List
- Audio/Visual Talks
- Other Doc Collections

Previous versions

Quick search

Enter search terms or a module, class or function name.

Python v2.6.1 documentation

Welcome! This is the documentation for Python 2.6.1, last updated Jan 29, 2009.

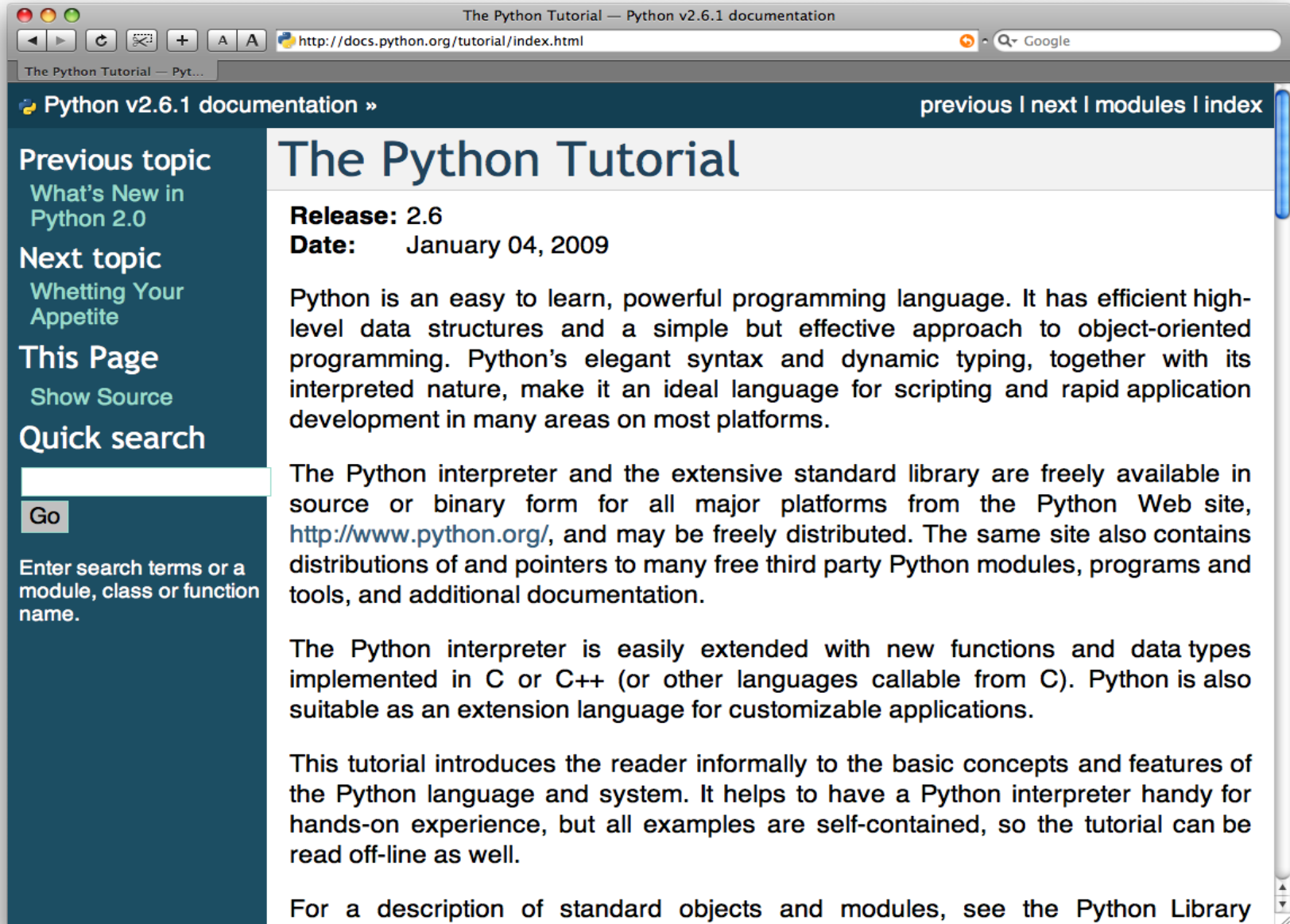
Parts of the documentation:

- What's new in Python 2.6?**
or all "What's new" documents since 2.0
- Tutorial**
start here
- Using Python**
how to use Python on different platforms
- Language Reference**
describes syntax and language elements
- Library Reference**
keep this under your pillow
- Python HOWTOs**
in-depth documents on specific topics
- Extending and Embedding**
tutorial for C/C++ programmers
- Python/C API**
reference for C/C++ programmers
- Installing Python Modules**
information for installers & sys-admins
- Distributing Python Modules**
sharing modules with others
- Documenting Python**
guide for documentation authors

Indices and tables:

- Global Module Index**
quick access to all modules
- General Index**
all functions, classes, terms
- Glossary**
the most important terms explained
- Search page**
search this documentation
- Complete Table of Contents**
lists all sections and subsections

The Python tutorial is good!



The screenshot shows a web browser window displaying the Python v2.6.1 documentation. The browser's address bar shows the URL <http://docs.python.org/tutorial/index.html>. The page title is "The Python Tutorial — Python v2.6.1 documentation". The page content is organized into a sidebar on the left and a main content area on the right. The sidebar contains links for "Previous topic" (What's New in Python 2.0), "Next topic" (Whetting Your Appetite), "This Page" (Show Source), and a "Quick search" box with a "Go" button. The main content area has a heading "The Python Tutorial" and a "Release: 2.6" and "Date: January 04, 2009" section. The main text describes Python as an easy-to-learn, powerful programming language with efficient high-level data structures and a simple but effective approach to object-oriented programming. It also mentions that the Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <http://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The text further states that the Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. The tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well. Finally, it mentions that for a description of standard objects and modules, see the Python Library.

The Python Tutorial — Python v2.6.1 documentation

http://docs.python.org/tutorial/index.html

Python v2.6.1 documentation » previous | next | modules | index

Previous topic
What's New in Python 2.0

Next topic
Whetting Your Appetite

This Page
Show Source

Quick search

Go

Enter search terms or a module, class or function name.

The Python Tutorial

Release: 2.6
Date: January 04, 2009

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.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <http://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

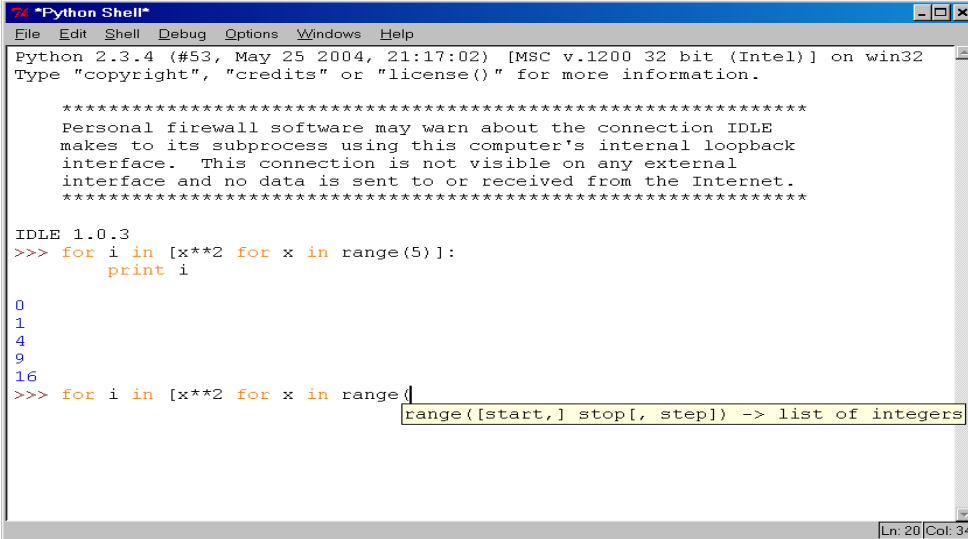
For a description of standard objects and modules, see the Python Library

Installing

- Download from <http://python.org/download/>
- There are several options for an IDE
 - IDLE – works well with Windows
 - Emacs with python-mode or your favorite text editor
 - Eclipse with Pydev (<http://pydev.sourceforge.net/>)

IDLE Development Environment

- IDLE is an Integrated DeveLopment Environment for Python, typically used on Windows
- Multi-window text editor with syntax highlighting, auto-completion, smart indent and other.
- Python shell with syntax highlighting.
- Integrated debugger with stepping, persistent breakpoints, and call stack visibility



```
Python 2.3.4 (#53, May 25 2004, 21:17:02) [MSC v.1200 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface.  This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 1.0.3
>>> for i in [x**2 for x in range(5)]:
    print i
0
1
4
9
16
>>> for i in [x**2 for x in range(5)]:
    range([start,] stop[, step]) -> list of integers
```

What is Python?

- Python is an
 - Open Source,
 - High Level,
 - Dynamically typed,
 - Compiled and
 - Interpreted,
 - Procedure Oriented and,
 - Object Oriented,
 - Generalized,
 - General Purpose,
 - Platform Independent,
 - Portable,
 - Extensible,
 - Robust
- Programming language.

Description

- **Open source:**
 - If the source code is available with machine code at the end user is known as open source. The benefit of open source is, we can modify the output by changing the source code at any moment.
- **High label:**
 - The label of a programming language is decided by the accessing style to memory by the programming language. Basically there are three labels are available,
 1. low label
 2. middle label
 3. high label
 - In every programming language, memory is compulsory used in back end. To handle the memory in backend, memory management mechanism is used. But in high label programming language that back end which is interacting with memory is hidden from the end user. In high label programming language all the memory management is the responsibility of programming language itself, not the responsibility of the programmer. End user is not responsible to write the memory management mechanism explicitly in high level programming language.

Contd.

[illegible]

The above program logic is a valid logic in C, C++, Java, and in Python, But this logic in C, C++ , Java will provide the output 0 for the factorial of 100. What the logic is written will work for C, and C++ up to a specific value, but for bigger number like 100 the same logic will not work. The problem is not present in logic, the problem is present in memory.

Contd.

- Statically typed Vs dynamically typed:**

C Program	C++ Program	Java Program	Python
<pre>#include "stdio.h" main () { int x ; x = 10 ; }</pre>	<pre>#include "iostream" using namespace std ; main () { int x ; x = 10 ; }</pre>	<pre>class demo { public static void main (String args []) { int x ; x = 10 ; } }</pre>	<pre>x = 10 print(x)</pre>

In C, C++, Java before using a variable, the declaration of data type is compulsory required (int x). If we will not provide the data type then it will generate an error i.e. variable is not declared. But in Python, the declaration of data type is not required.

Before using a variable if it is compulsory to declare the data type of that variable is known as statically typed programming language. If it is not required to specify the data type of a variable before using that is called dynamically typed programming language. So Python is a dynamically typed programming language.

Contd.

• Procedure Oriented Vs Object Oriented:

C Program	Java Program
<pre>#include "stdio.h" main () { function (); } void function () { // implement logic here }</pre>	<pre>class demo { public static void main (String args []) { method (); } void method () { // implement logic here } }</pre>

- According to Procedure oriented, to implement the logic, function is required. That function may be main function or may be any other function. Function is also known as procedure.
- But according to object oriented to implement the logic, method is used. A method is nothing but a function present inside the class. To access, a method caller object is required of same class. In object oriented if we will write a function without a class that is an error.

Contd.

Python Program

```
def function () :  
    # implement logic here  
    pass  
  
class demo :  
    def method ( self ) :  
        # implement logic here  
        pass
```

But in python we can implement the logic using function also by using method. So we can say python is both Procedure oriented and object oriented programming language.

Contd.

- **General purposed Programming Language:**
 - If a programming language can be used different types of application, then that programming language is known as general purpose programming language. The different types of applications are:
 - Desktop Application
 - Gaming Application
 - Web based Application
 - Embedded System Application
 - Robotics Application
 - Real Time System
 - Database Application

Contd.

- **Generalized programming language**
 - If a single function can work with different types of parameter that function is known as generalized function and that language is known as generalized programming language.

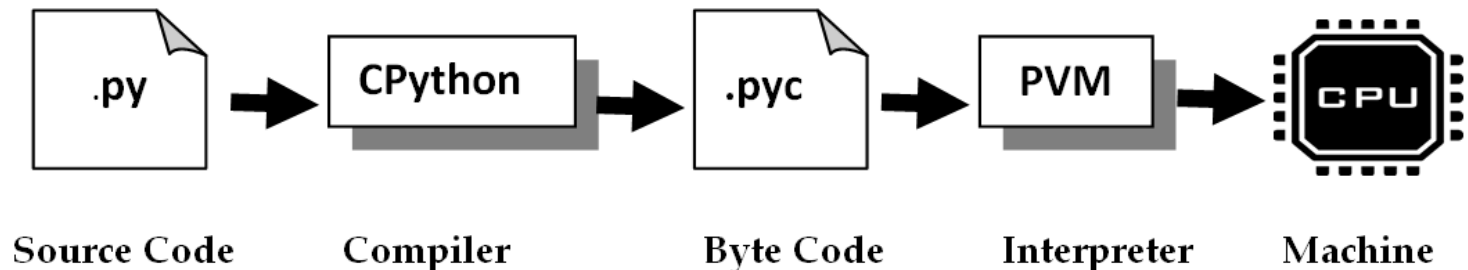
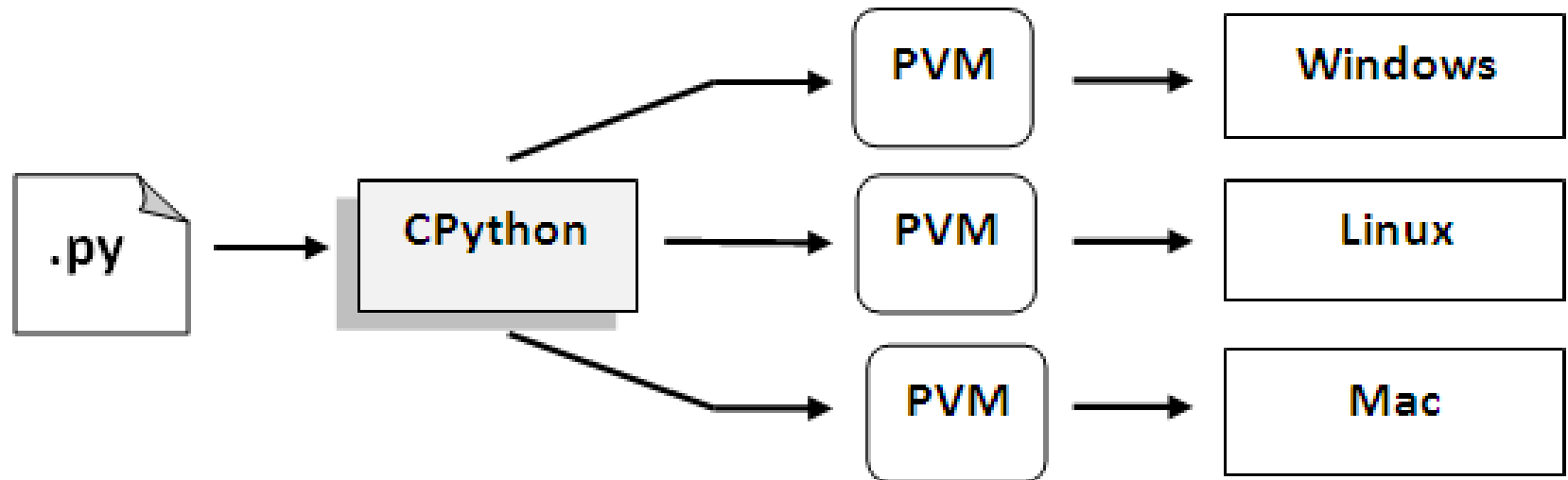
C Program	Python Program
<pre>#include "stdio.h" using namespace std ; void myprint(int x) { printf ("%d \n", x); } main () { myprint (10); myprint (11.22); myprint ('a'); myprint ("India"); }</pre>	<pre>def myprint (x) : print x myprint (10) myprint (11.22) myprint ('a') myprint ("India")</pre>

Contd.

- In that program of C, there is an user defined function `myprint` to print an integer value. If we will pass float or char or string instead of an integer there will be no any error in compilation but the output will not come properly for float, double, char and string. For float/ double instead of 11.22, the output will come 11, for char instead of ``a`` the output will come ASCII value of ``a`` i.e 97, and for string instead of “india” the memory address of base element will come.
- But in Python program, for every types of value belongs to any data types, that single function can provide the output according to the type is given to that function.
- So we can say python is also a generalized programming language.

Contd.

- Platform Independent



Contd.

- **Portable**

- Portability means moving the instruction of one language from one operating environment to another operating environment. It allows developing programs irrespective of hardware.
- It is not required to be confused between a platform-independent programming language and portable programming language. Portability is used in the domain of hardware (Micro Processor 32bit/ 64 bit) whereas platform independent used in the domain of operation system.

Contd.

- **Extensible**

- A program is written in C or C++ can be integrated into Python or CPython and can be executed using PVM. To integrate Java code with Python code and run on JVM, Jython is used. And to integrate .NET code and library with Python code, Iron Python is useful that will run on CLR (Common Language Runtime).

- **Robust**

- Nothing is more important than to recognize error for a programmer in his program.
- When some errors come in python, the interpreter provides a “stack trace” with use full information such as, at which position the program is crashed and why, with the file name, line number and reason.

Contd.

```
[root@localhost demo]# vi test.py
```

```
x = 10
```

```
print 'The value of x is :', x
```

```
print 'The value of y is :', y
```

```
print 'Bye'
```

```
[root@localhost demo]# python test.py
```

```
The value of x is : 10
```

```
The value of y is :
```

```
Traceback (most recent call last):
```

```
  File "test.py", line 5, in <module>
```

```
    print 'The value of y is :', y
```

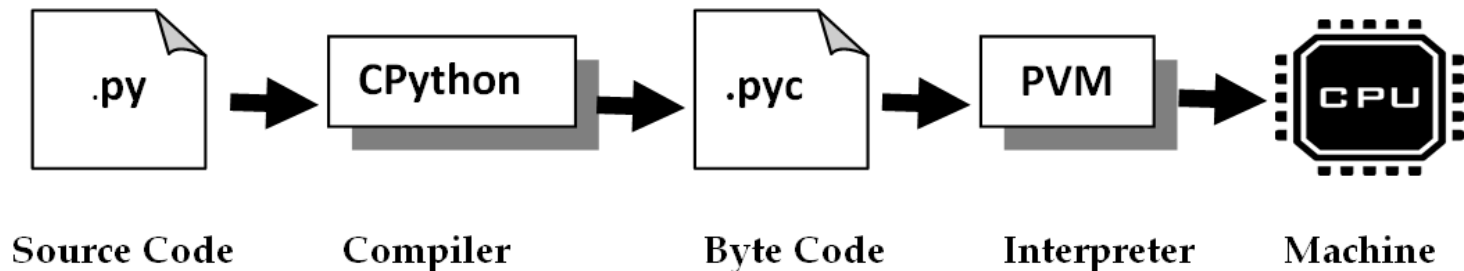
```
NameError: name 'y' is not defined
```

Contd.

- In the above program, the variable x is defined but y is not defined, so x is a variable in my program but y is not a variable in my program. That's why when the value of y is printed; the process is terminated without printing Bye message in the output screen and error information is provided, like the file name is test.py, the line number is 5, in the same module, and the reason that name `y` is not defined i.e. NameError in python.

How Python Works?

- Python is basically called as Interpreted language but in Python only interpreter is not working alone. In Python internally two steps are working, the first step is the Compiler and the second step is the Interpreter.
 - Compiler (CPython)
 - Interpreter (PVM)



Contd.

- What the code we are writing in .py file is the python source code.
- The source code will be converted to Byte Code with the help of CPython compiler.
- The Byte code extension in python is .pyc. pyc stands for python compiled file.
- What the byte code is created that will be executed by the virtual machine of python i.e. PVM (Python Virtual Machine).

Script mode and Interactive mode

- To write a program in python there are two different modes are available.
 - Script mode
 - Interactive mode
- In script mode, we will create a file and then write the program in that file having the extension .py. But in the interactive mode, we have to start the python IDLE. Interactive mode is basically used to test small code. What the code is written in interactive mode, that code will be not saved. We can start the python interactive mode by ``python`` command in command prompt or in the terminal if python is installed. By ``python`` command python2 version will be started and to start python 3 version interactive mode ``python3`` command is used.

Summary

- The source code extension in python is .py.
- Python is basically called interpreted programming language. But in python internally two steps are working.
 - compiler
 - interpreter
- The standard compiler in python is CPython.
- The interpreter of python is technically called as Python Virtual Machine(PVM).
- Like java in python byte code is created after compilation.
- The byte code extension in python is .pyc.
- pyc stands for python compiled file.
- CPython compiler is implemented using C language.
- JPython compiler is implemented using Java language.
- In JPython compiler, byte code extension is .class which can be executed by JVM.
- To write a program in python, there are two different modes are available.
 - script mode
 - interactive mode