1. **Introduction**

**Topics:**

* 1. About Python Page 1
  2. Features of Python Page 2
  3. Applications for Python Page 3,4
  4. Environment, Installation ,Basic Usage Page 5-11
  5. Running Python Page 12
  6. **About Python :**

.C 🡪 .obj -🡪 .exe

.java 🡺 .class -> jar file, ear, war file

.py Interpreter **cpython jython ironpython** .py file, Interpreter

1. Source file –compilation🡪 2. Compiled file 🡪 3 .exe/jar file

.c .java .obj .class ready to execute

.py - compiliation -> .pyc 🡪 Execution

Python was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python is a general-purpose **interpreted, interactive, object-oriented, and high-level** programming language.

Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

**Features:**

* **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.

## History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

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

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

* 1. **Features of Python :**
* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the programmer 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.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a **scripting language** or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports **dynamic type checking. Dynamically typed programming language.**

int x = 10 # JAVA, others…

**x = 10** # Python

* It supports **automatic garbage collection.**
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.
  1. **Applications for Python : Web application Developer**

Python is used in many application domains. Here's a sampling.

The Python Package Index lists thousands of third party modules for Python.

1. **Web and Internet Development**

Python offers many choices for web development:

* Frameworks such as **Django** and **Pyramid**.
* Micro-frameworks such as **Flask** and **Bottle**.
* Advanced content management systems such as Plone and django CMS.

Python's standard library supports many Internet protocols:

* **HTML and XML**
* **JSON**
* E-mail processing.
* Support for FTP, IMAP, and other Internet protocols.
* Easy-to-use socket interface.

And the Package Index has yet more libraries:

* **Requests**, a powerful HTTP client library.
* BeautifulSoup, an HTML parser that can handle all sorts of oddball HTML.
* Feedparser for parsing RSS/Atom feeds.
* Paramiko, implementing the SSH2 protocol.
* Twisted Python, a framework for asynchronous network programming.

1. **Scientific and Numeric :**

Python is widely used in scientific and numeric computing:

* SciPy is a collection of packages for mathematics, science, and engineering.
* Pandas is a data analysis and modeling library.
* IPython is a powerful interactive shell that features easy editing and recording of a work session, and supports visualizations and parallel computing.
* The Software Carpentry Course teaches basic skills for scientific computing, running bootcamps and providing open-access teaching materials.

1. **Education**

Python is a superb language for teaching programming, both at the introductory level and in more advanced courses.

* Books such as How to Think Like a Computer Scientist, Python Programming: An Introduction to Computer Science, and Practical Programming.
* The Education Special Interest Group is a good place to discuss teaching issues.

1. **Desktop GUIs**

The Tk GUI library is included with most binary distributions of Python.

Some toolkits that are usable on several platforms are available separately:

* wxWidgets
* Kivy, for writing multitouch applications.
* Qt via pyqt or pyside

Platform-specific toolkits are also available:

* GTK+
* Microsoft Foundation Classes through the win32 extensions

1. **Software Development**

Python is often used as a support language for software developers, for build control and management, testing, and in many other ways.

* SCons for build control.
* Buildbot and Apache Gump for automated continuous compilation and testing.
* Roundup or Trac for bug tracking and project management.

1. **Business Applications**

Python is also used to build ERP and e-commerce systems:

* Odoo is an all-in-one management software that offers a range of business applications that form a complete suite of enterprise management applications.
* Tryton is a three-tier high-level general purpose application platform.
  1. **Environment, Installation, Basic Usage :**

Python is available on a wide variety of platforms including **Windows, Linux and Mac OS**

## Local Environment Setup

Open a terminal window and type "python" to find out if it is already installed and which version is installed.

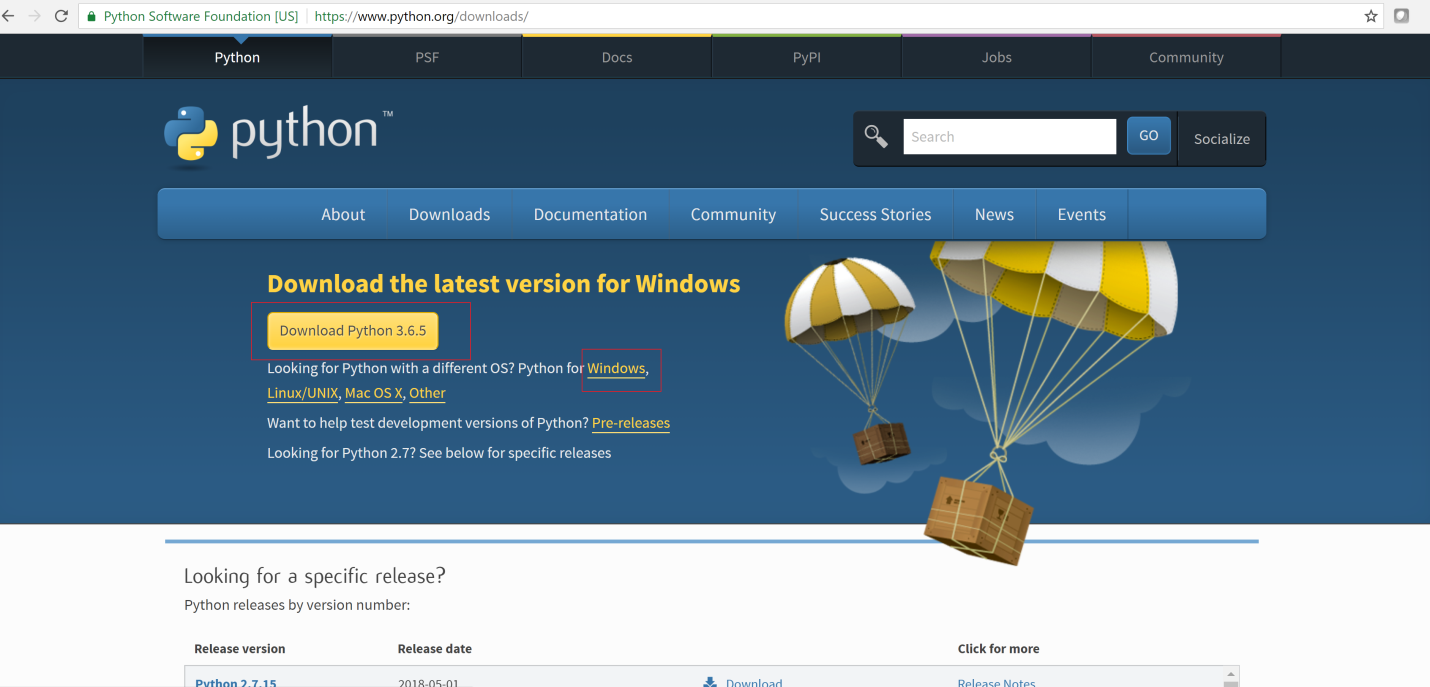
* Unix (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX, etc.)
* Win 9x/NT/2000/XP/BVista/2007/10
* Macintosh (Intel, PPC, 68K)
* OS/2
* DOS (multiple versions)
* PalmOS
* Nokia mobile phones
* Windows CE
* Acorn/RISC OS
* BeOS
* Amiga
* VMS/OpenVMS
* QNX
* VxWorks
* Psion
* Python has also been ported to the Java and .NET virtual machines

## Getting Python

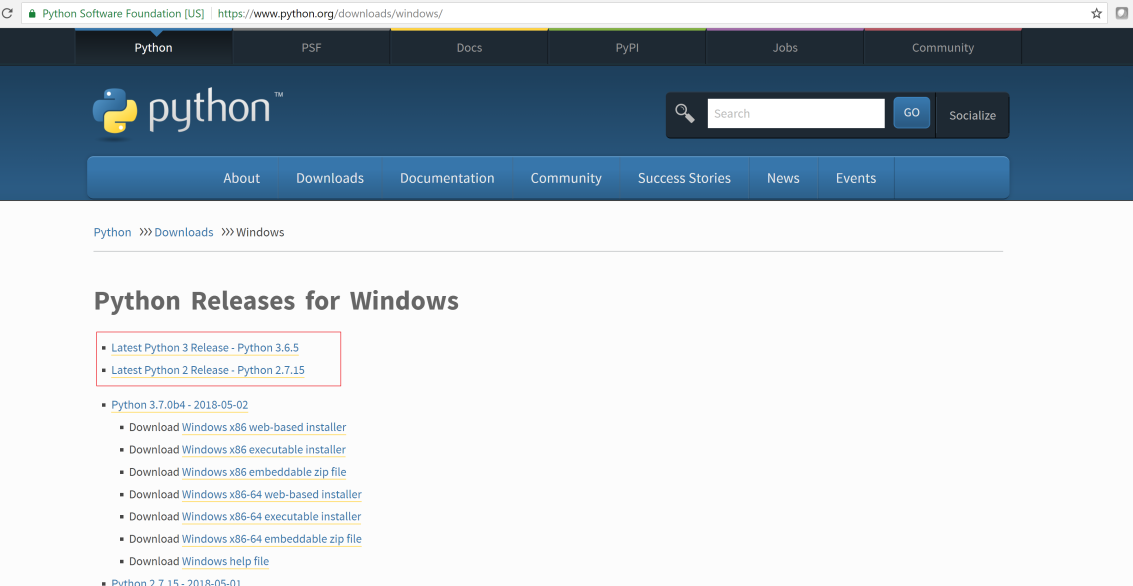
* The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org/>
* You can download Python documentation from <https://www.python.org/doc/>. The documentation is available in HTML, PDF, and PostScript formats.

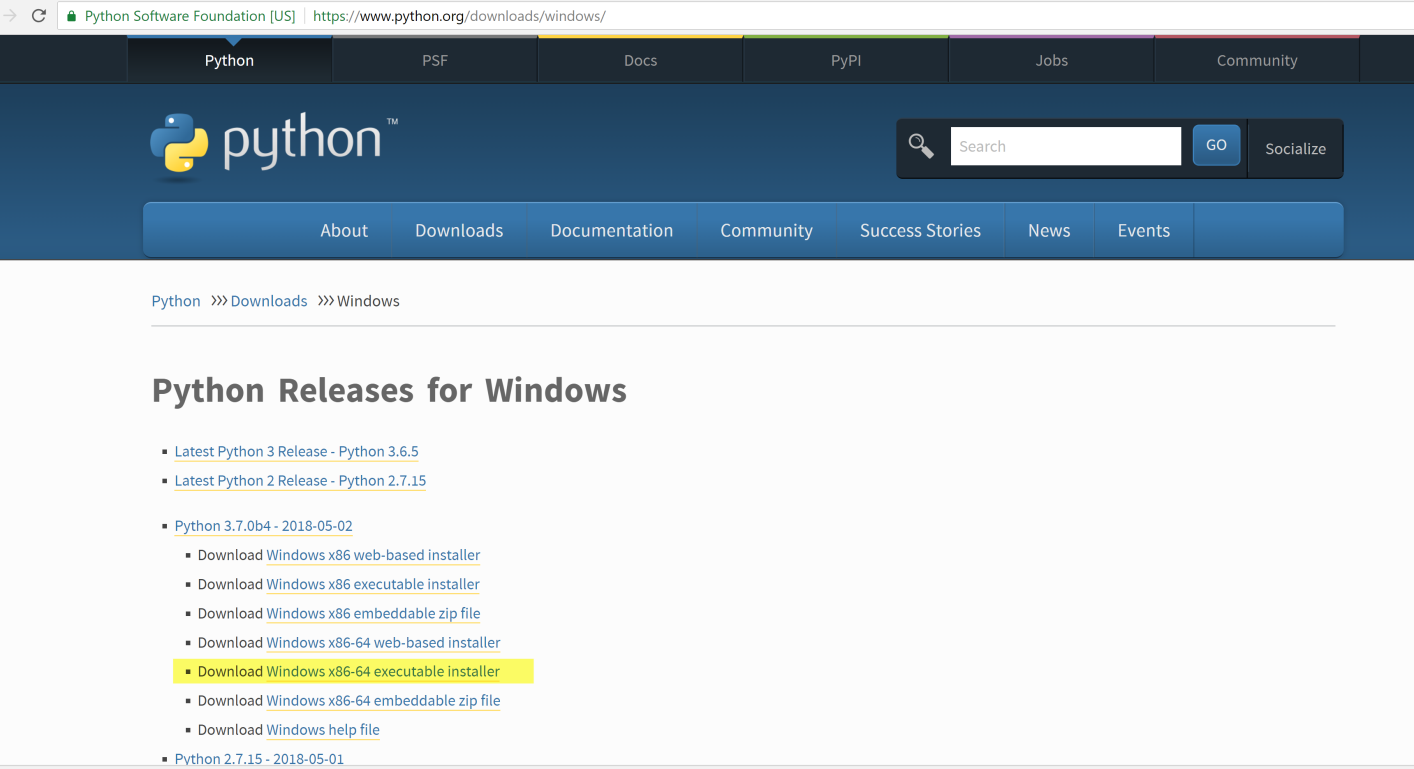
2.X Vs 3.X 2.7 vs 3.9

**Step 1 :** Go to python.org and select appropriate file and download it. **3.8**



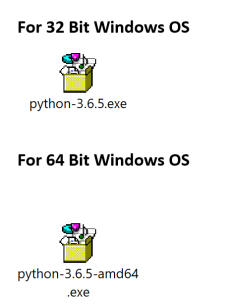
**Step 2 :** Select 3.X for latest version,2.X for older version





**Step 3:** Download above highlighted version for windows **64 Bit OS**,”**Windows x86 executable installer”** for **32 Bit OS,**

If it is for Linux, download Linux compatible version.

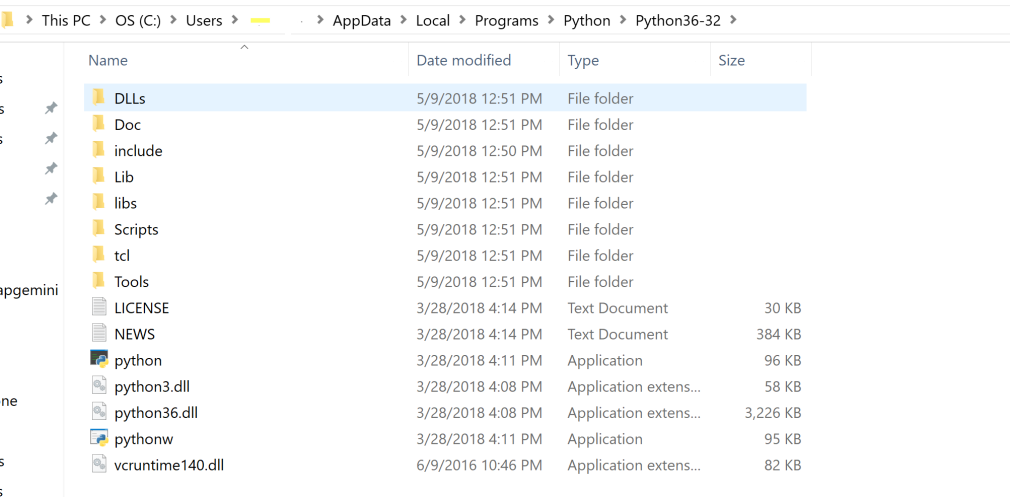


**Step 4 :** Double click on above file, and finish installation. You can install anywhere in your computer.

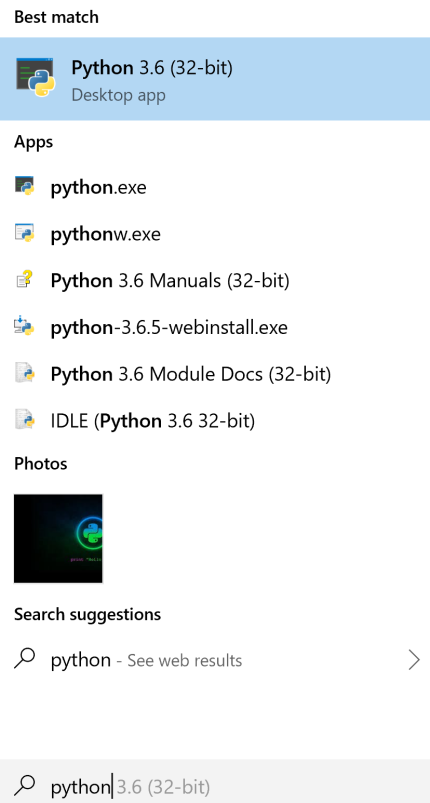
After installation by default you can find python software at below location.

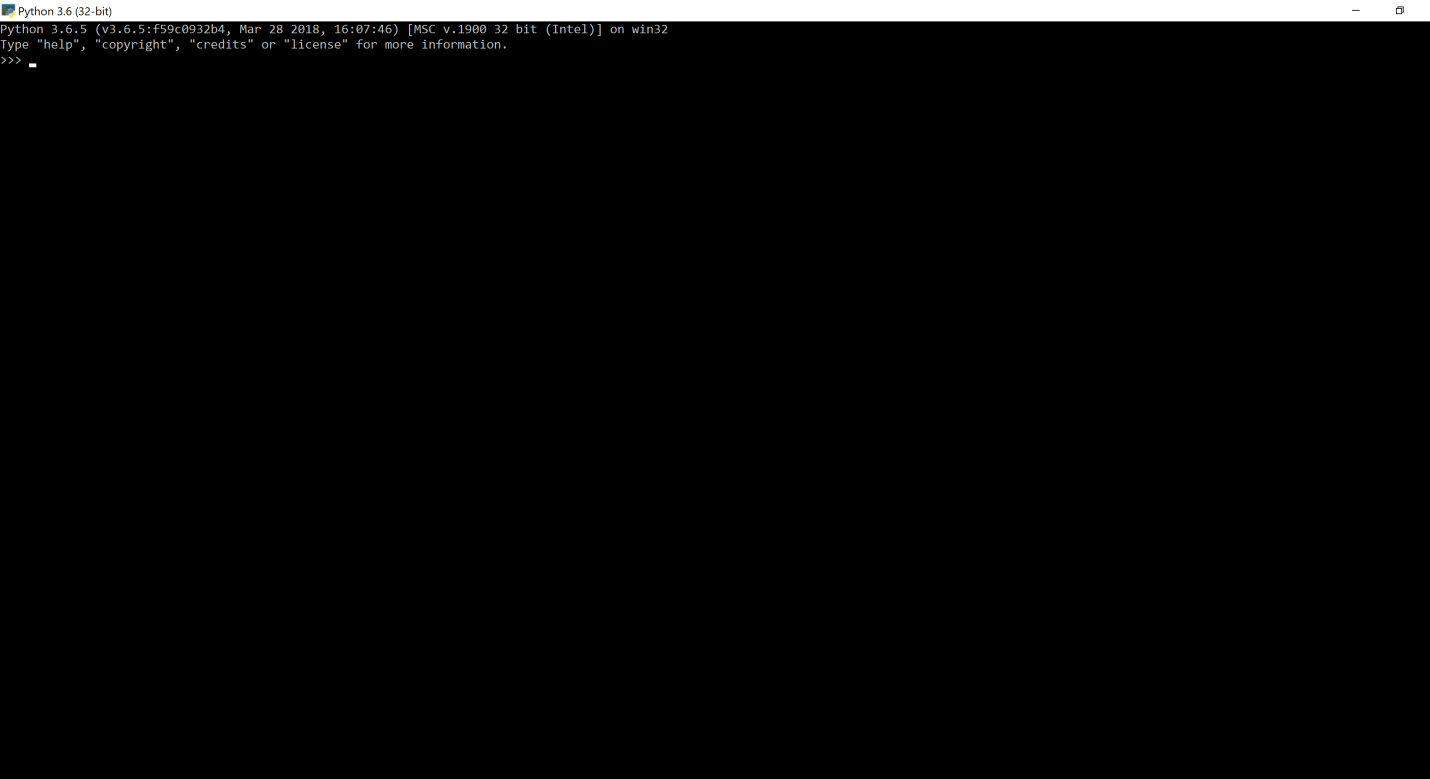
C:\Users\<user>\AppData\Local\Programs\Python\Python36-32\ => 32 Bit

**C:\Users\<user>\AppData\Local\Programs\Python\Python36-64\** => 64 Bit



**Step 5 :**Go to Search bar and type python. Click on Python 3.6(32 bit).You can create shortcut on desktop.

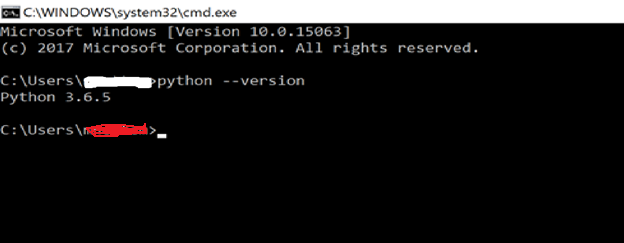




To check python installed correctly (OR) to check version of python in our system

Windows + R -> cmd and then click on Enter

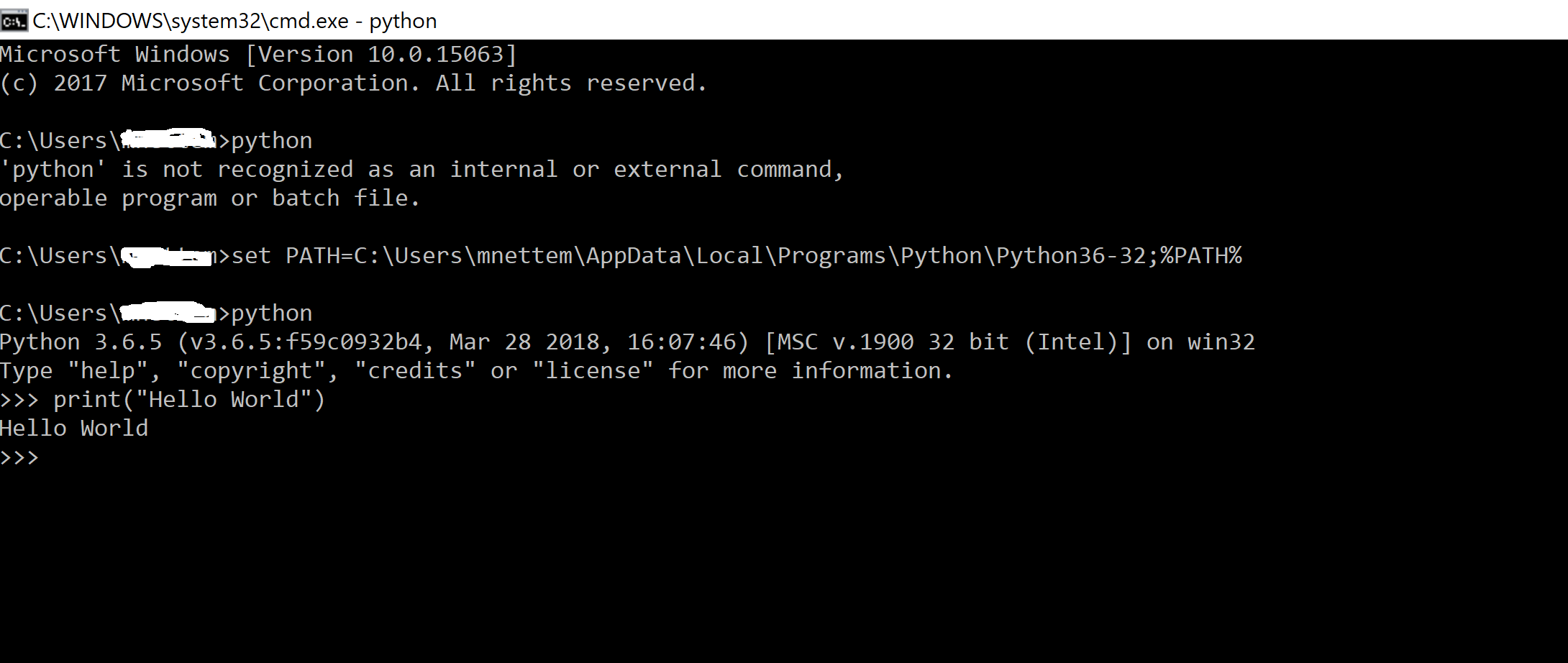
Give command **python --version**



## Setting path at Windows

To add the Python directory to the path for a particular session in Windows At the

**1. command prompt** **(Temporary)**−

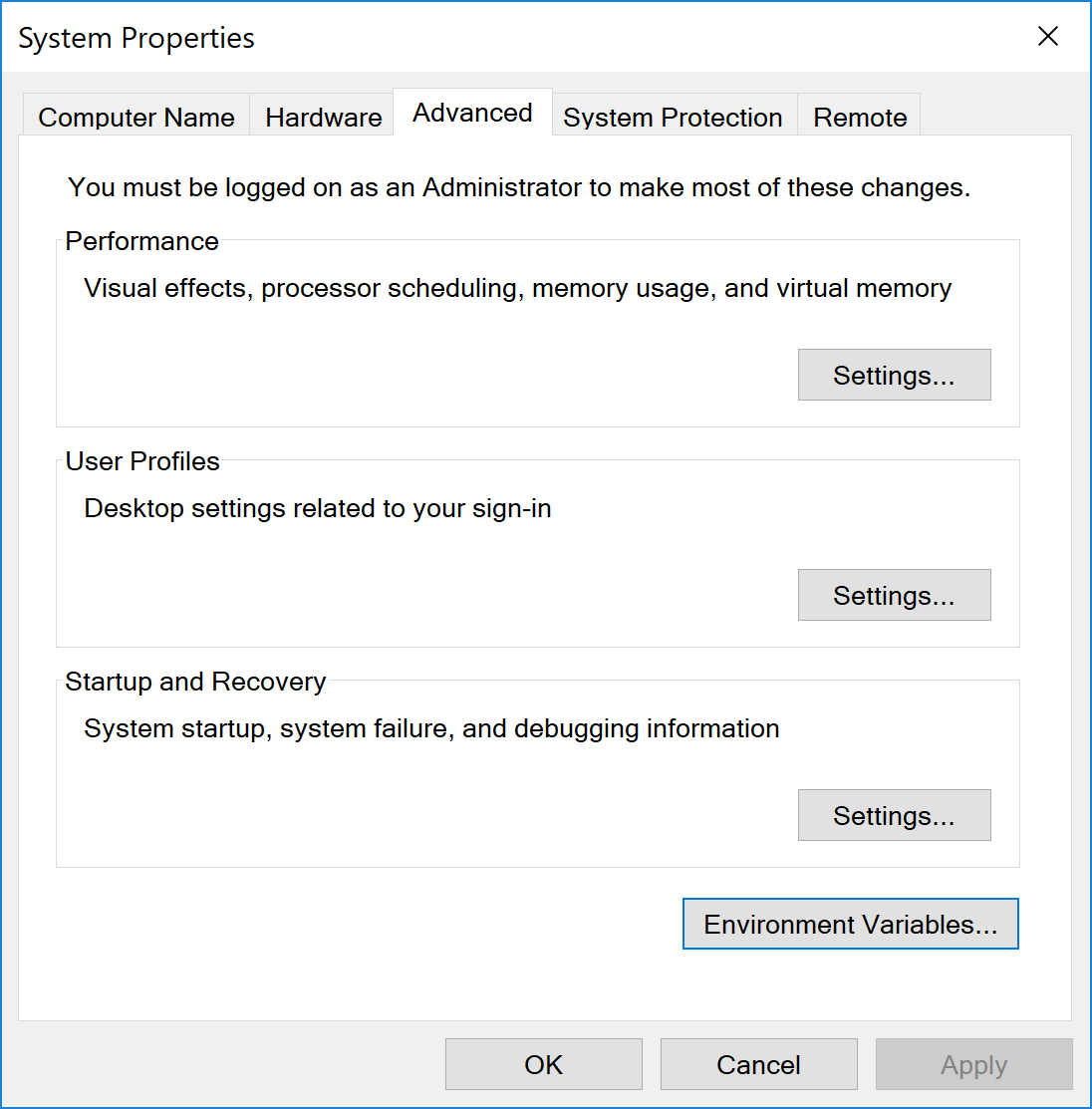


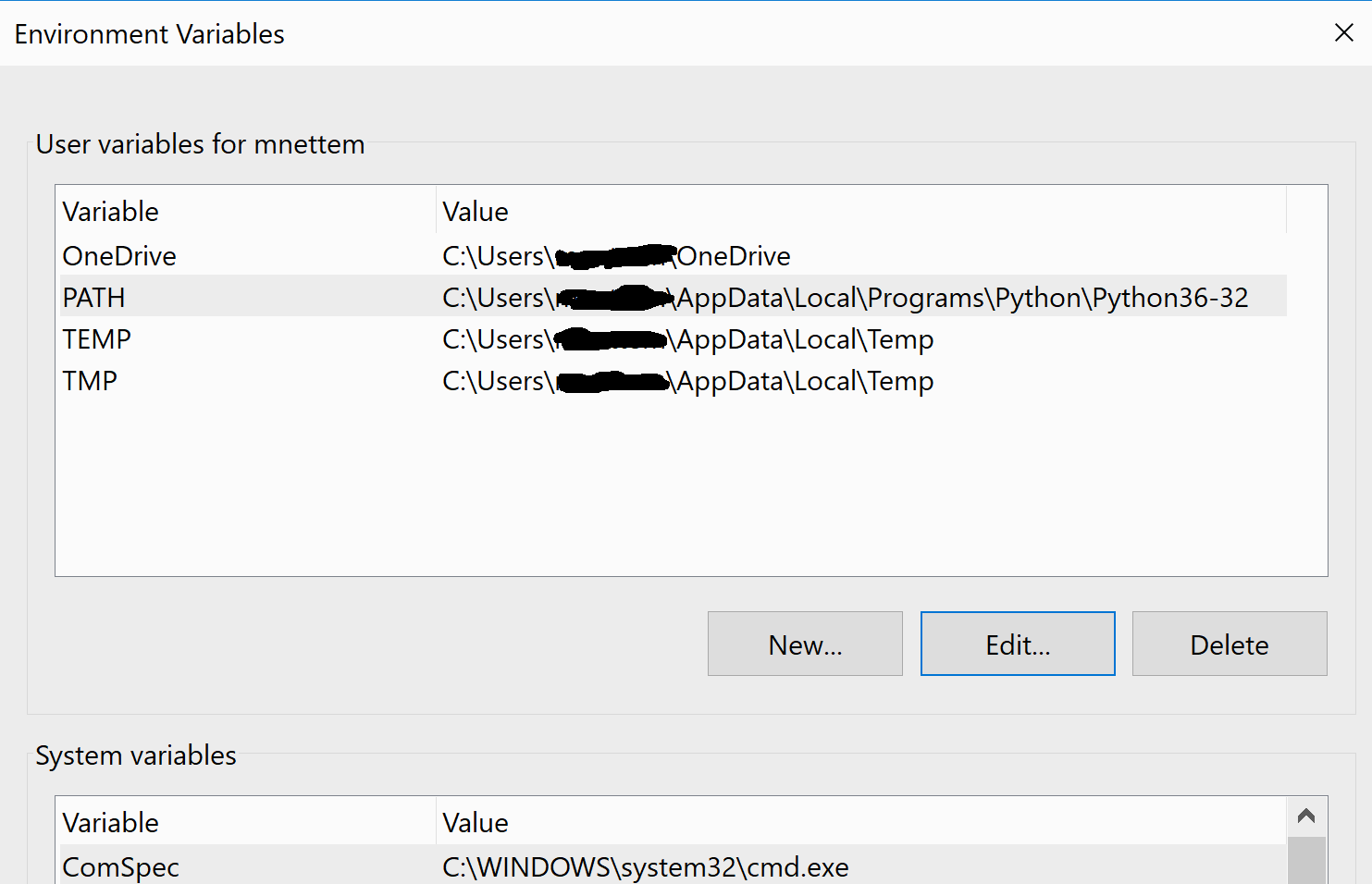
**2. Environmental Variable Setup(Permanent) :**

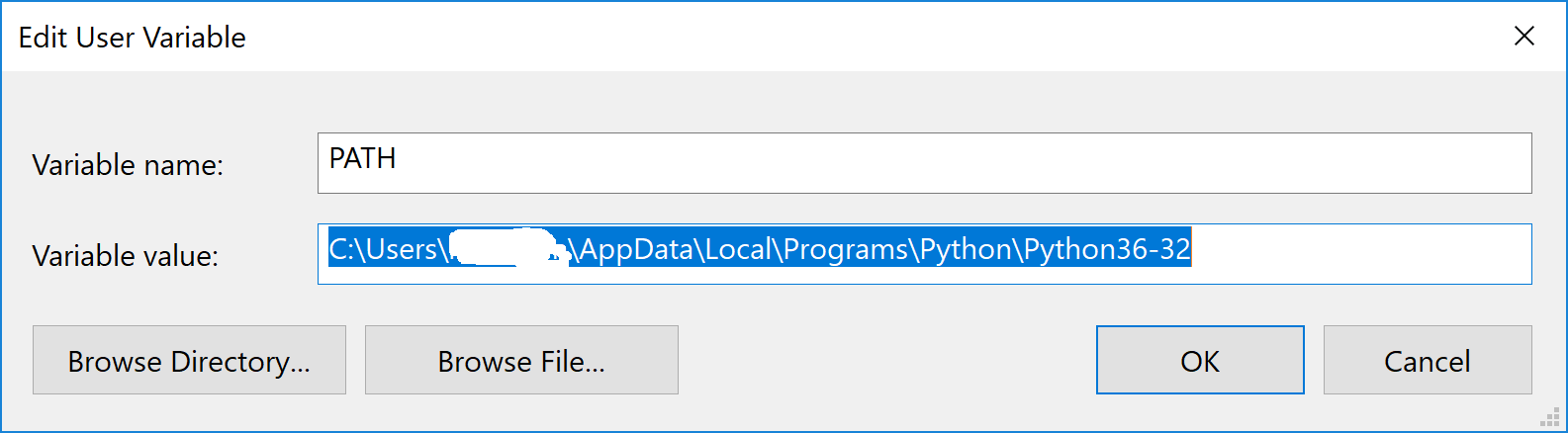
Go to

My Computer -> Properties -> Advanced System Settings ->Environment Variables -> User Variables ->

**Path :**

**C:\Users\<user name>\AppData\Local\Programs\Python\Python36-32\**

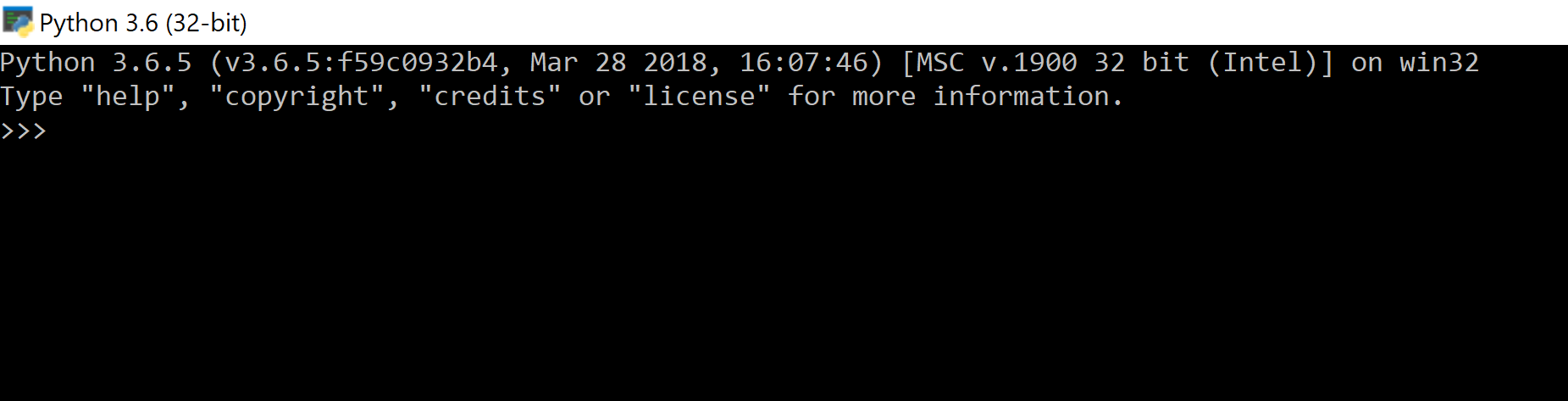




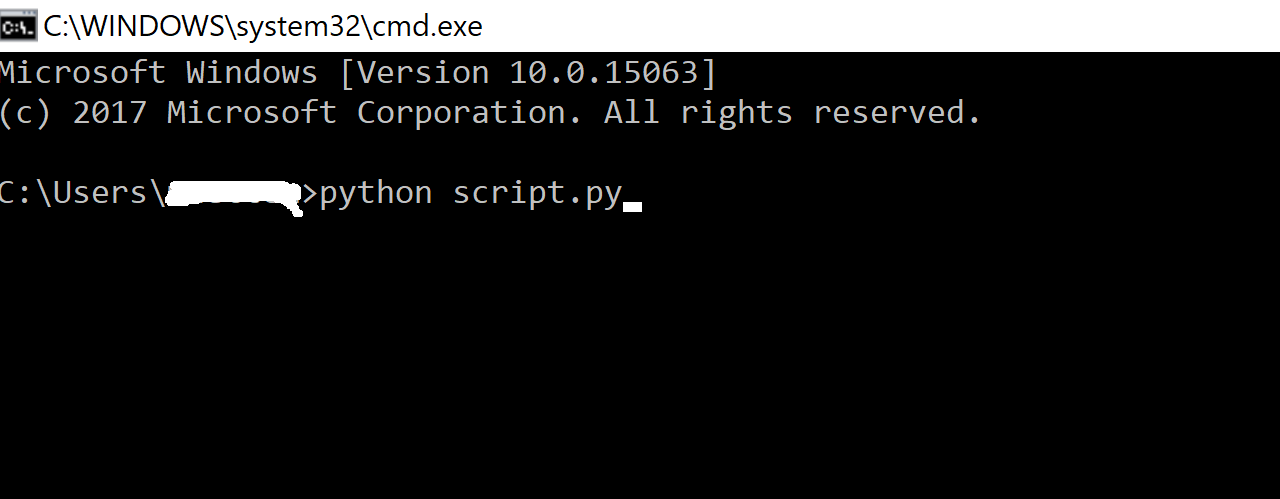
**1.4 Running Python :**

There are three different ways to start Python –

### **1. Interactive Interpreter :**



### **2. Script from the Command-line:**



### **3. Integrated Development Environment:**

You can run Python from a Graphical User Interface (GUI) environment as well, if you have a GUI application on your system that supports Python.

* **Unix** − IDLE is the very first Unix IDE for Python.
* **Windows** − PythonWin is the first Windows interface for Python and is an IDE with a GUI.

For both Windows,Linux you can use different IDE’s like **Eclipse, PyCharm** etc.,

**Important Questions:**

**1. Introduction :**

How interpreter works in python.

Python features

Interpreter CPython IronPython Jython

Dynamically typed programming language

Procedure Oriented vs Object oriented programming languages

Setting path in python. Importance

Environment variables importance in python

Different ways to run python program

*source file  
compiler  
interpreter  
interpreted\* programming language  
cpython vs ironpyhon vs jython:  
interactive  
object-oriented  
high-level programming language  
functional programming and structured programming  
procedural vs object-oriented programming  
Dynamically typed programming language int x = 10 <==> x = 10  
Automatic garbage collection*