1. **Introduction**

**Topics:**

* 1. About Python Page 1
  2. Features of Python Page 4
  3. Internals of Python Page 6
  4. Environment, Installation, Basic Usage Page 10
  5. Running Python Page 16
  6. **About Python:**

**Python** is a general purpose, **dynamic**, [high-level](https://www.javatpoint.com/classification-of-programming-languages), and **interpreted** 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).

**General Purpose:**  Is a programming language that can create all types of programs. **Dynamic\*\*:** The types and values are both dynamic, which means the types and values can both be changed. For example, a variable that was previously assigned an integer can be assigned a string. The type checking is done during run time. **High Level:** A high-level language is one that is user-oriented in that it has been designed to make it straightforward for a programmer to convert an algorithm into program code. A low-level language is machine oriented. Low-level programs are expressed in terms of the machine operations that must be performed to carry out a task.

Below are some facts about Python Programming Language:

* Python is currently the most widely used multi-purpose, high-level programming language.
* Python allows programming in Object-Oriented and Procedural paradigms.
* Python programs generally are smaller than other programming languages like Java. Programmers must type relatively less and indentation requirement of the language, makes them readable all the time.
* Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
* The biggest strength of Python is huge collection of standard libraries which can be used for the following:
  + Web frameworks like [**Django**](https://www.geeksforgeeks.org/django-tutorial/)**\*\*, Flask\*\*, FAST API\*, Pyramid, Bottle** (used by YouTube, Instagram, Dropbox)
  + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like [Kivy](https://www.geeksforgeeks.org/kivy-tutorial/" \t "_blank), Tkinter, PyQt etc., )
  + Image processing (like [OpenCV](https://www.geeksforgeeks.org/opencv-python-tutorial/), Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia
  + Scientific computing
  + Text processing and many more

**Prerequisites: Nothing with common sense**

### **Reason for increasing popularity:**

1. Emphasis on **code readability, shorter codes**, ease of writing
2. Programmers can express logical concepts in **fewer lines**of code in comparison to languages such as C++ or Java.
3. Python supports **multiple** programming paradigms, like object-oriented, imperative and functional programming or procedural.
4. There exist inbuilt functions for almost all of the frequently used concepts.
5. Philosophy is “Simplicity is the best”

**Python vs Java**

| **Python** | **Java** |
| --- | --- |
| **Dynamically Typed: x = 10**   * No need to declare anything. An assignment statement binds a name to an object, and the object can be of any type. * No **type casting** is required when using container objects | **Statically Typed int x = 10**   * All variable names (along with their types) must be explicitly declared. Attempting to assign an object of the wrong type to a variable name triggers a type of exception. * Type casting is required when using container objects. |
| Concise Express much in limited words | Verbose Contains more words |
| Compact | Less Compact |
| Uses **Indentation** for structuring code | Uses **braces** for structuring code |

Java Code:

**public** **class** HelloWorld

{

**public** **static** **void** main(String[] args)

   {

      System.out.println("Hello, world!");

   }

}

Python Code:

print("Hello, world!")

**Pros:**

1. Ease of use
2. Multi-paradigm Approach

**Cons:**

1. Slow speed of execution compared to C, C++
2. Absence from mobile computing and browsers
3. For the C, C++ programmers switching to python can be irritating as the language requires proper indentation of code. Certain variable names commonly used like sum are functions in python. So, C, C++ programmers must look out for these.

**Organizations using Python:**

1. Google (Components of Google spider and Search Engine)
2. Yahoo (Maps)
3. YouTube
4. Mozilla
5. Dropbox
6. Microsoft
7. Cisco
8. Spotify
9. Quora

* 1. **Features of Python:**

**a. Interpreted\*\***: **.C .obj .exe .java .class .war/.ear .py**

* There are no separate compilation and execution steps like C and C++, Java
* Directly run the program from the source code.
* Internally, Python converts the source code into an intermediate form called bytecodes then translated into native language of specific computer to run it.
* No need to worry about linking and loading with libraries, etc.

**b. Platform Independent:**

* Python programs can be developed and executed on multiple operating system platforms.
* Python can be used on Linux, Windows, Macintosh, Solaris and many more.

**C. Free and Open Source:**Redistributable

d. **High-level Language**:

In Python, no need to take care about low-level details such as managing the memory used by the program.

**e. Simple:**

* + Closer to learning Mathematics basics in English medium. Easy to Learn
  + More emphasis on the solution to the problem rather than the syntax

**f. Embeddable**

* + Python can be used within C/C++ program to give scripting capabilities for the program’s users.

**g. Robust:**

* + Exceptional handling features
  + Memory management techniques in built

**h. Rich Library Support**

* + The Python Standard Library is very vast.
  + Known as the “**batteries included**” philosophy of Python. It can help do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.

**i. Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

**j. Easy-to-read** − Python code is more clearly defined and visible to the eyes.

**k. Easy-to-maintain** − Python's source code is fairly easy-to-maintained

**l. Databases** − Python provides interfaces to all major commercial databases.

**Applications of Python:**

1 GUI based desktop applications

1. Web frameworks and applications
2. Enterprise and Business applications
3. Operating Systems
4. Education
5. Database Access
6. Language Development
7. Prototyping
8. Software Development

10 Graphic design, image processing applications, Games, and Scientific/ computational Applications

**1.3 Internals of Python:**

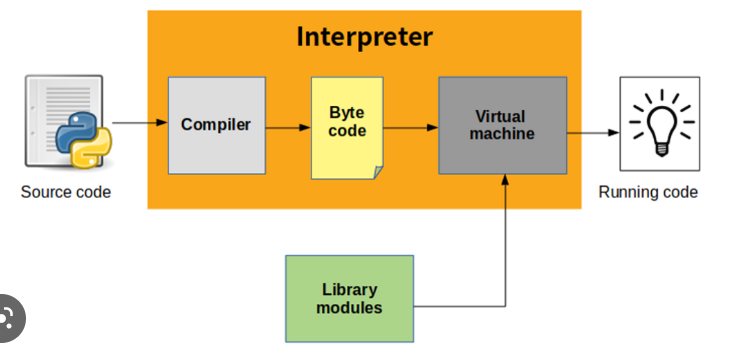
**PYHTON CODE (.py file)**

**Syntax Checker and Translator**

**BYTE CODE**

**PVM (Python Virtual Machine)**

**OUTPUT**

****

**Python**is an object-oriented programming language like Java.

Python is called an interpreted language.

Python uses code modules that are interchangeable instead of a single long list of instructions that was standard for functional programming languages.

The standard implementation of python is called “**cpython**”. It is the default and widely used implementation of Python.   
Python doesn’t convert its code into machine code, something that hardware can understand. It converts it into something called byte code.

So within python, compilation happens, but it’s just not into a machine language.

It is into byte code (.pyc or .pyo) and this byte code can’t be understood by the CPU. So we need an interpreter called the python virtual machine to execute the byte code

**Step 1:**

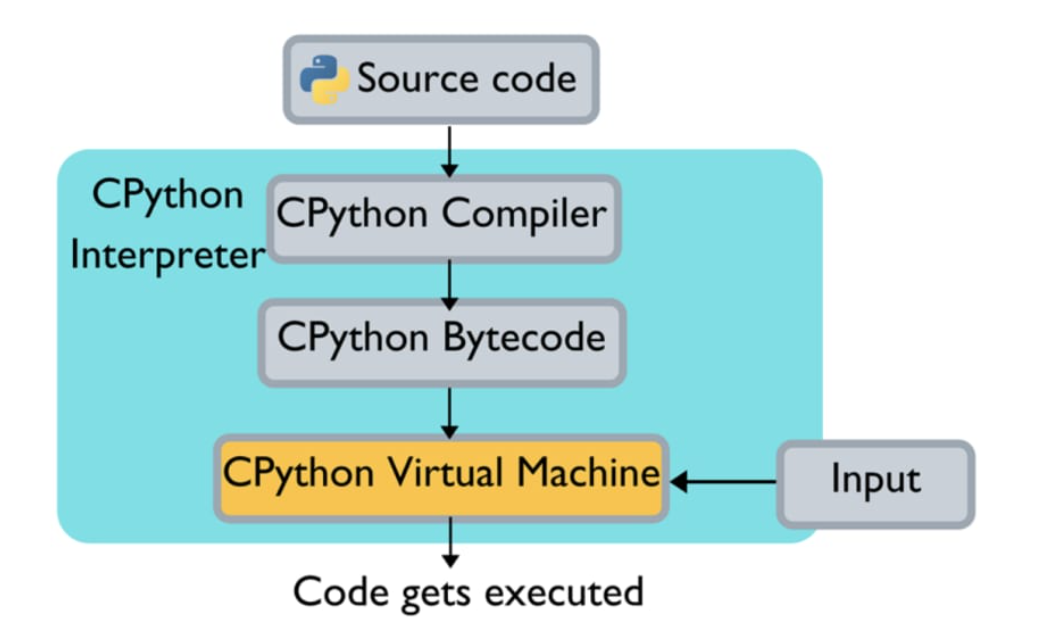
The python compiler reads a python source code or instruction. Then it verifies that the instruction is well-formatted, i.e., it checks the syntax of each line. If it encounters an error, it immediately halts the translation and shows an error message.

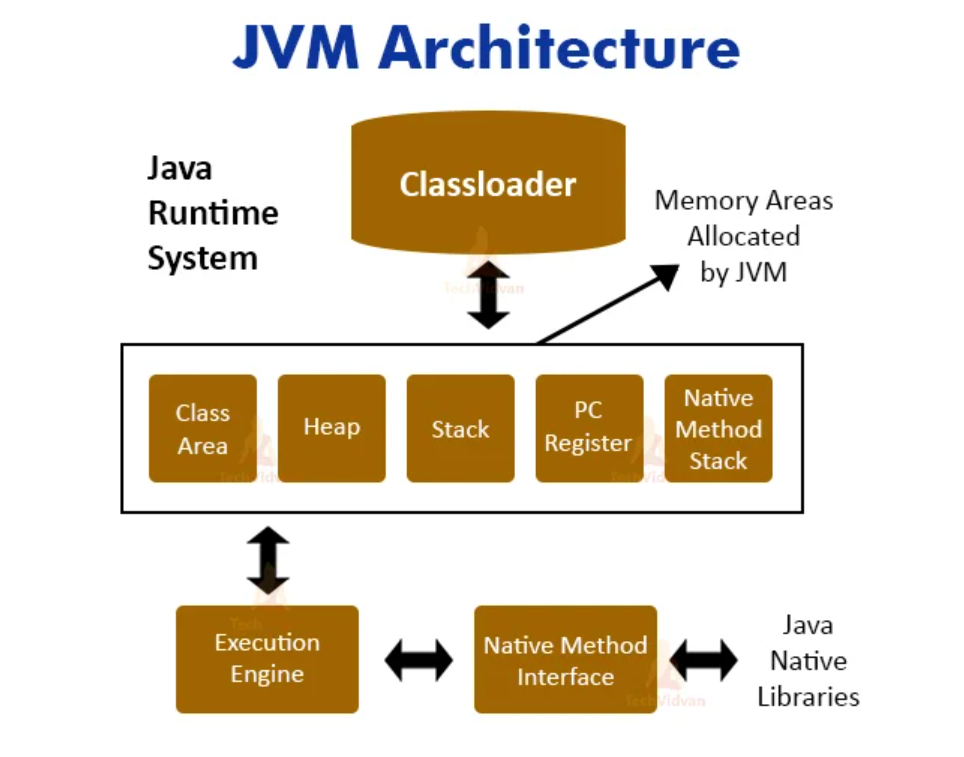
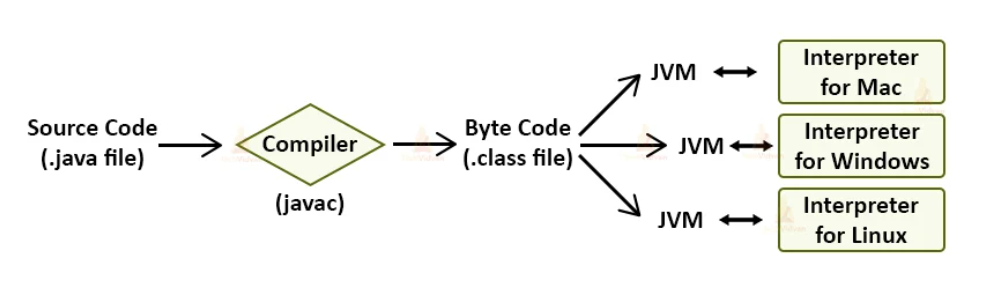
**Step 2:**

 If there is no error, i.e., if the python instruction or source code is well-formatted then the compiler translates it into its equivalent form in an intermediate language called “Byte code”.

**Step 3:**

Byte code is then sent to the Python Virtual Machine (PVM) which is the python interpreter. PVM converts the python byte code into machine-executable code. If an error occurs during this interpretation, then the conversion is halted with an error message.



****

***Python’s Unique Feature:***

**# Python is Dynamically Typed Programming Language**

* It can be used as a **scripting language** or can be compiled to bytecode for building large applications.
* It provides very high-level dynamic data types and supports **dynamic type checking.**

**# JAVA, others…**

**int x = 10; float x = 10; (5L <– 2L)**

**int x = 10 ;**

**type |variable |Operator | Value ;**

**# Python:**

**x = 10** **y = 10.5**

Maths: Get value of expression 2x2+3x+1 when x = 10

STATE : x = 10

BEHAVIOR : 2x2+3x+1

As given x = 10

Substitute x value in given expression

= 2x2+3x+1

= 2(10)(10)+3(10)+1

= 200+30+1

= 231

**1.4 Environment, Installation, Basic Usage of Python:**

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.

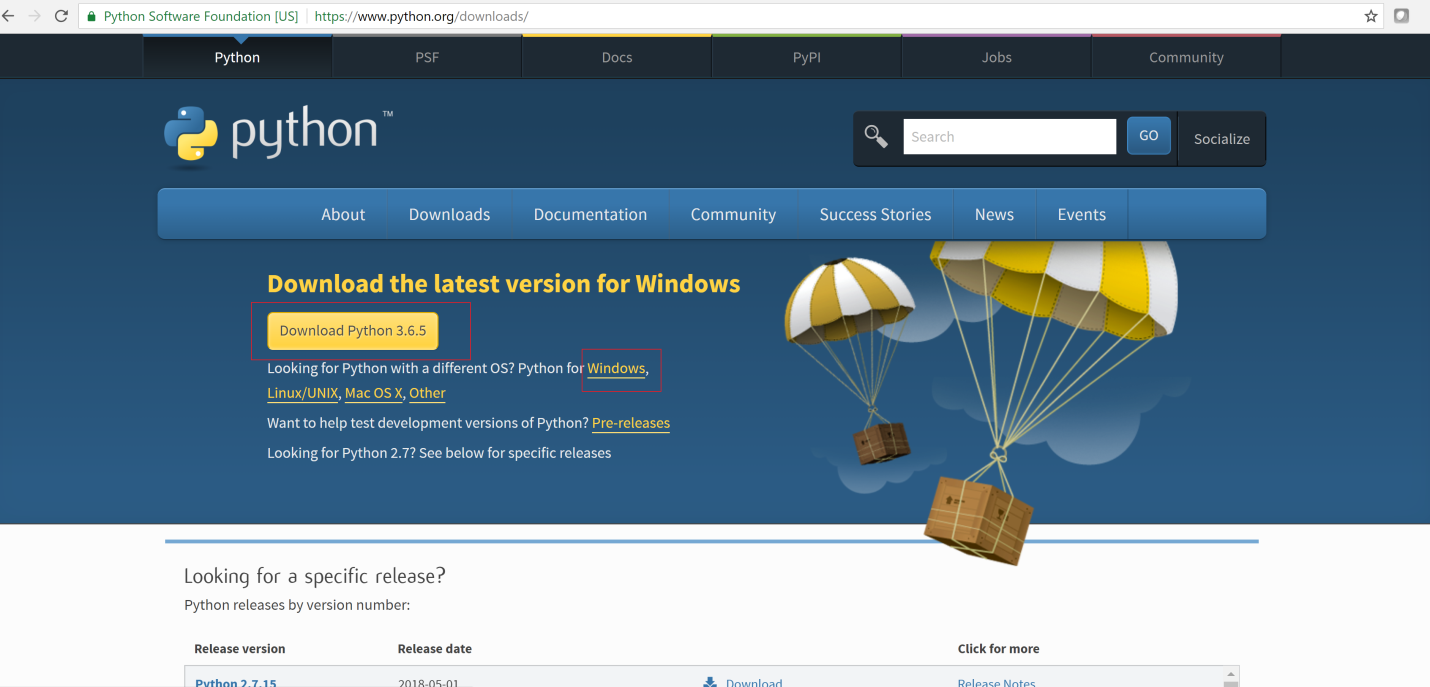
* Unix (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX, etc.)
* Win 9x/NT/2000/XP/BVista/2007/10
* Macintosh (Intel, PPC, 68K)
* OS/2
* DOS (multiple versions)
* PalmOS
* Nokia mobile phones
* Windows CE
* Acorn/RISC OS
* BeOS
* Amiga
* VMS/OpenVMS
* QNX
* VxWorks
* Psion
* Python has also been ported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

## Getting Python

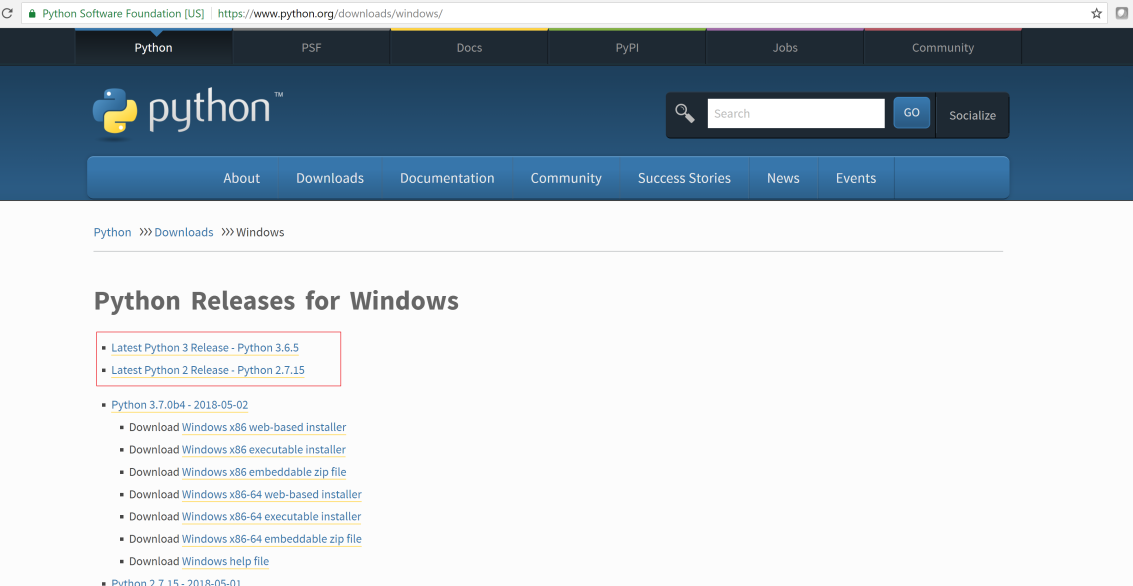
* The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python [**https://www.python.org/**](https://www.python.org/)
* You can download Python documentation from <https://www.python.org/doc/>. The documentation is available in HTML,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

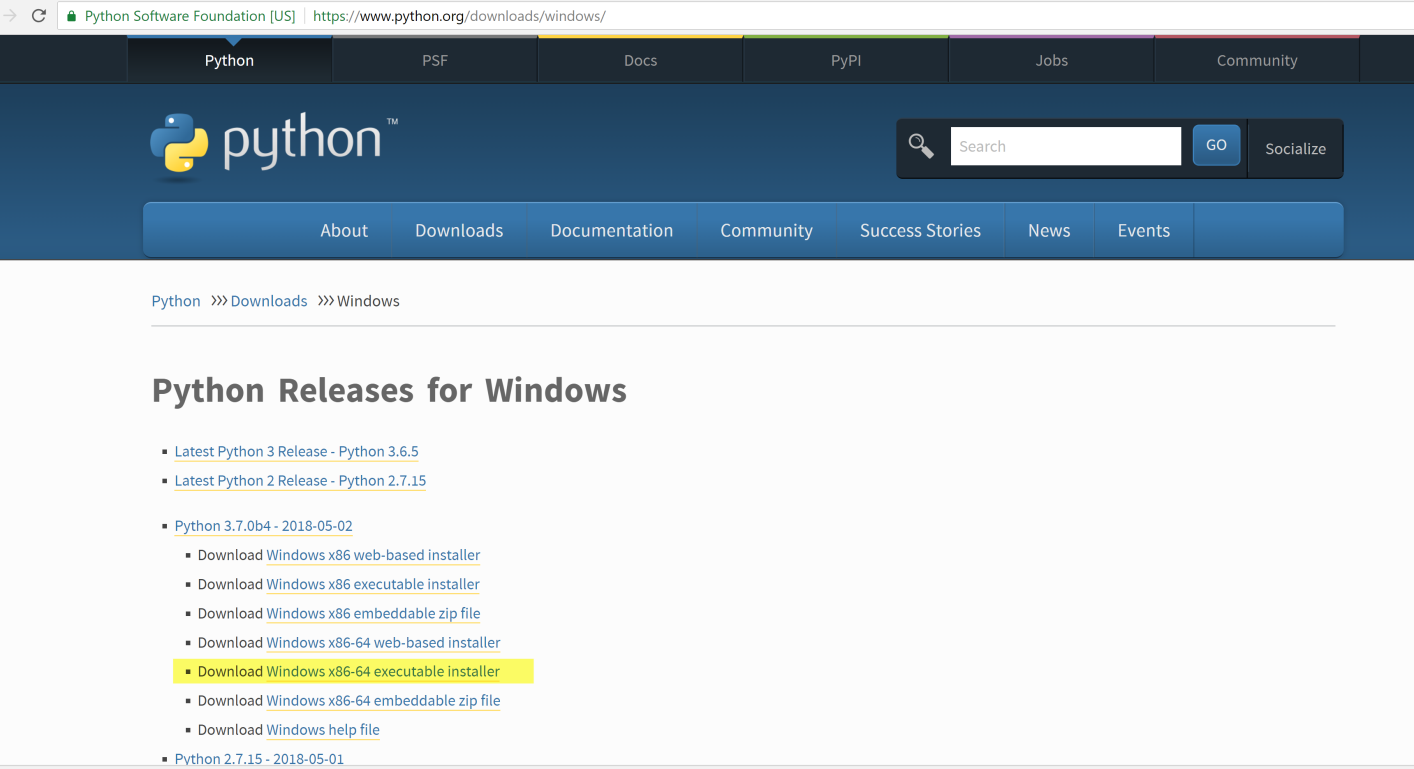
2.X Vs 3.X

**Step 1 :** Go to python.org and select appropriate file and download it. **3.7/3.8**



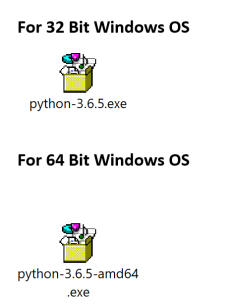
**Step 2 :** Select 3.X for latest version,2.X for older version





**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.



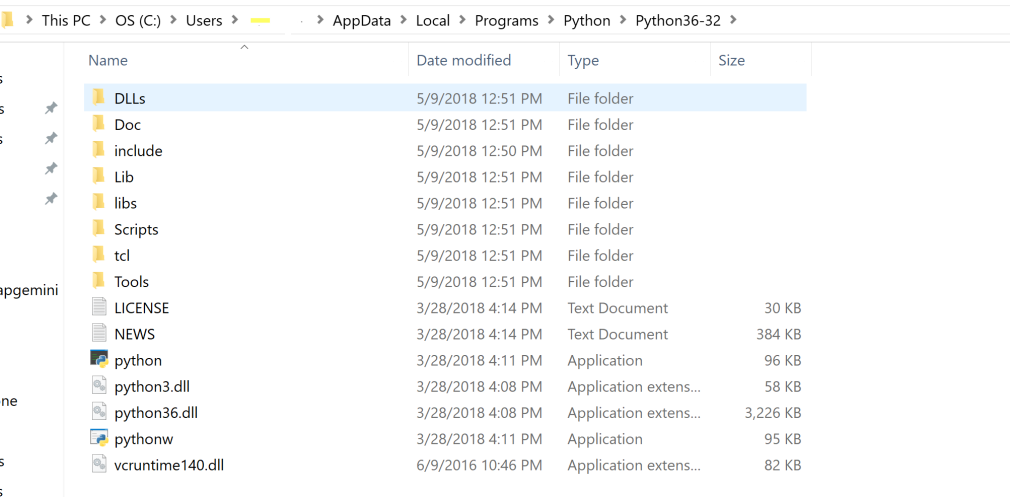
**Step 4 :** Double click on above file and finish installation. You can install anywhere in your computer.

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

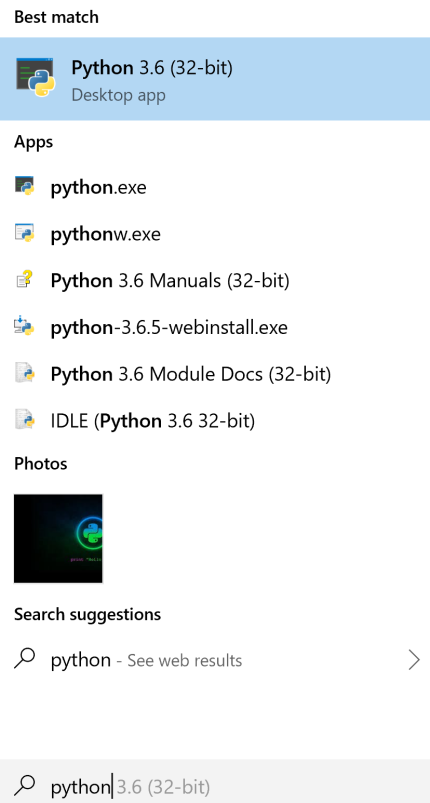
C:\Users\**<user>**\AppData\Local\Programs

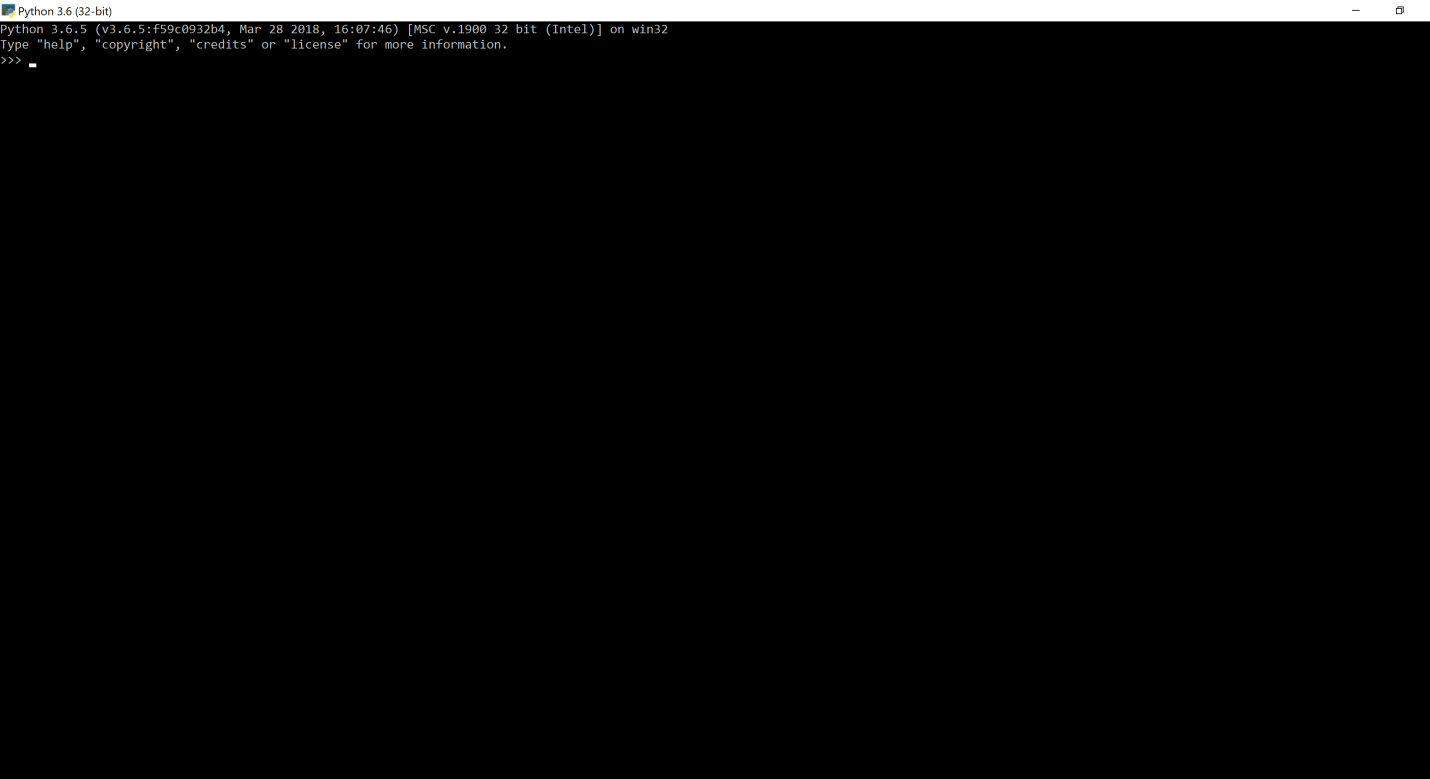
=> 32 Bit

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



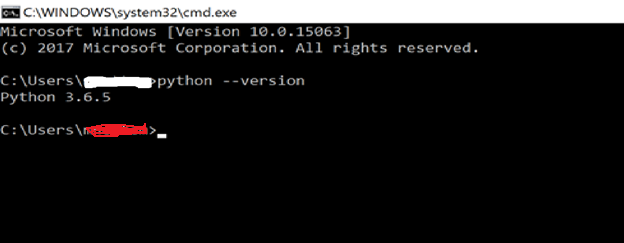
**Step 5:** Go to Search bar and type python. Click on Python 3.6(32 bit).You can create shortcut on desktop.





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

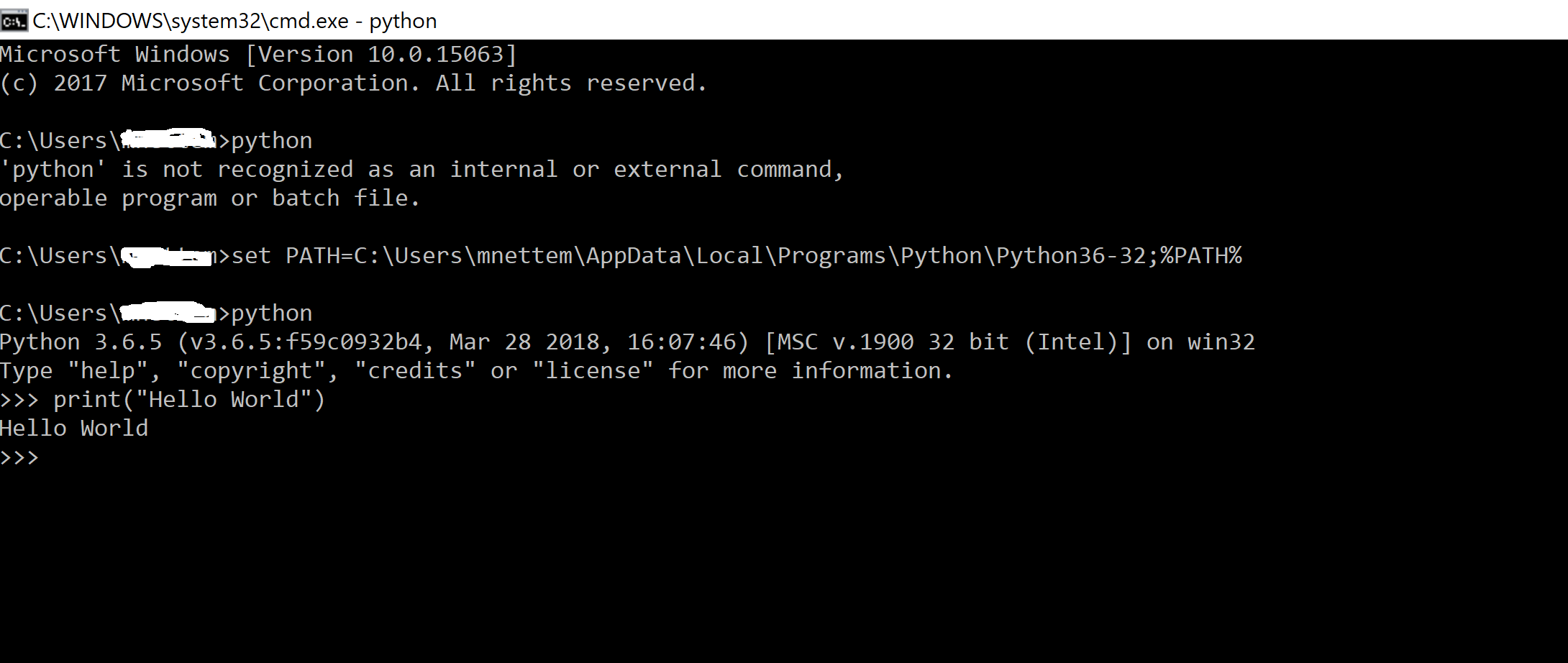
Give command **python –version**

Pyhton 

## Setting path at Windows

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

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

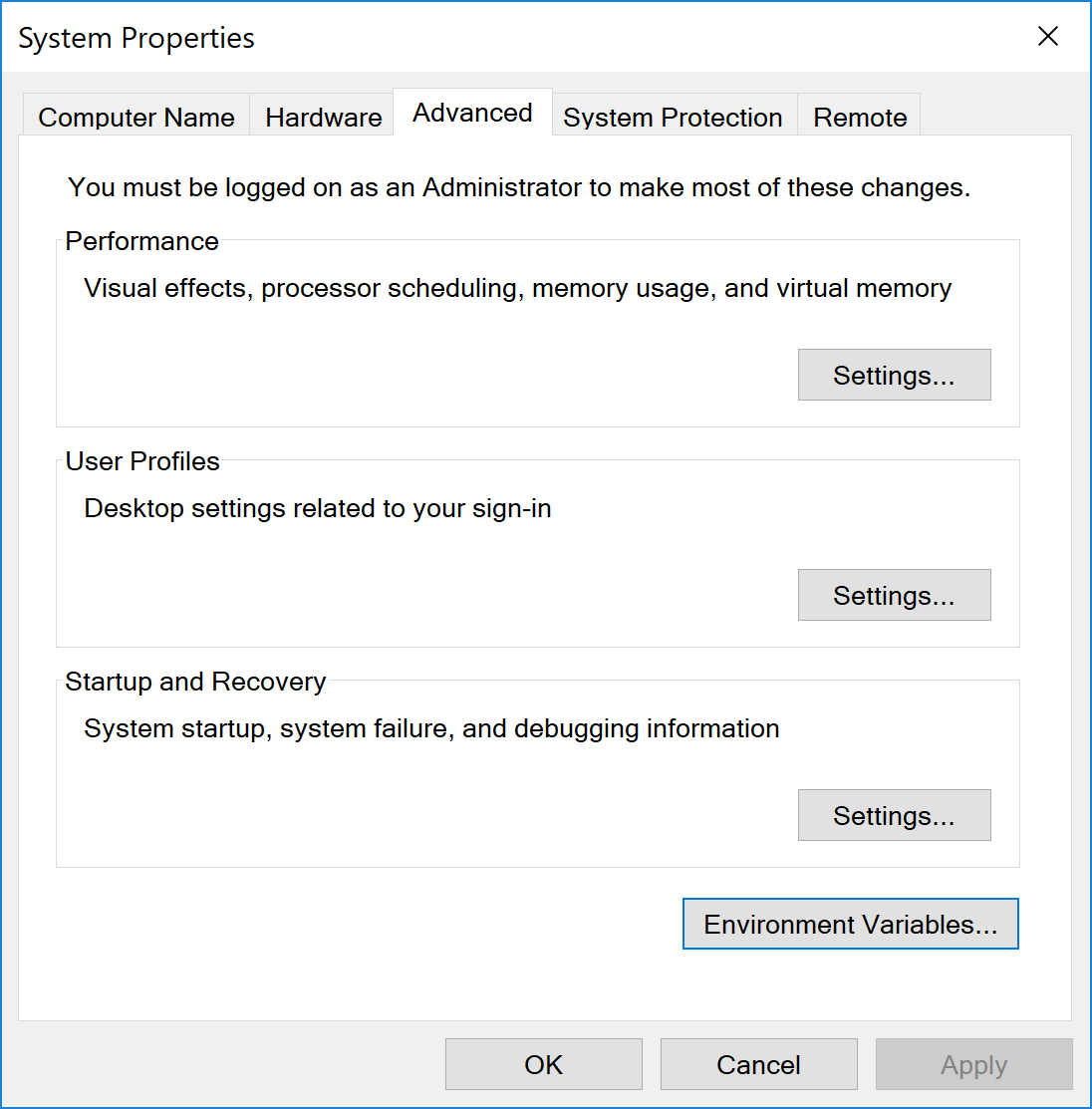


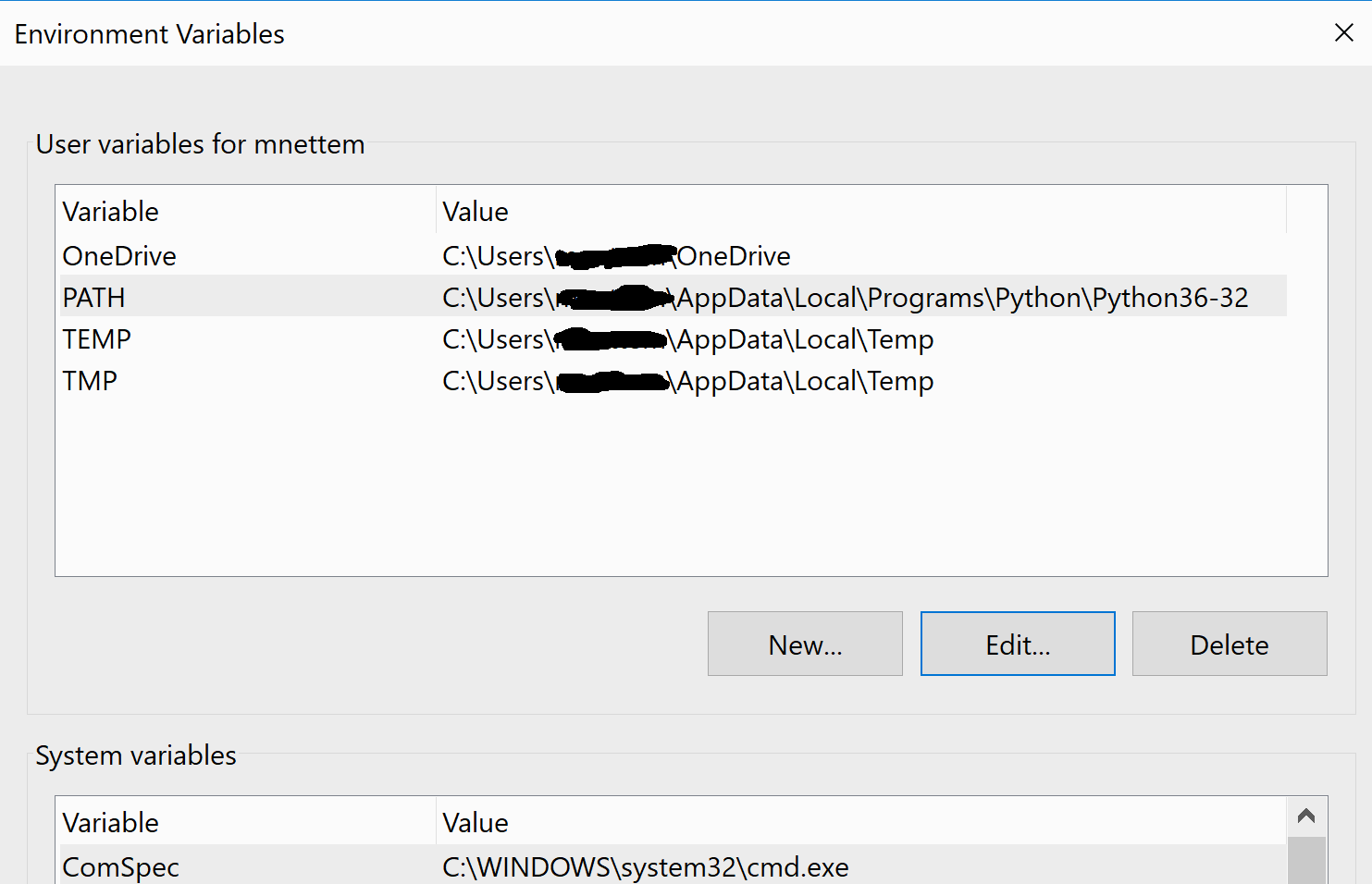
**2. Environmental Variable Setup (Permanent) :**

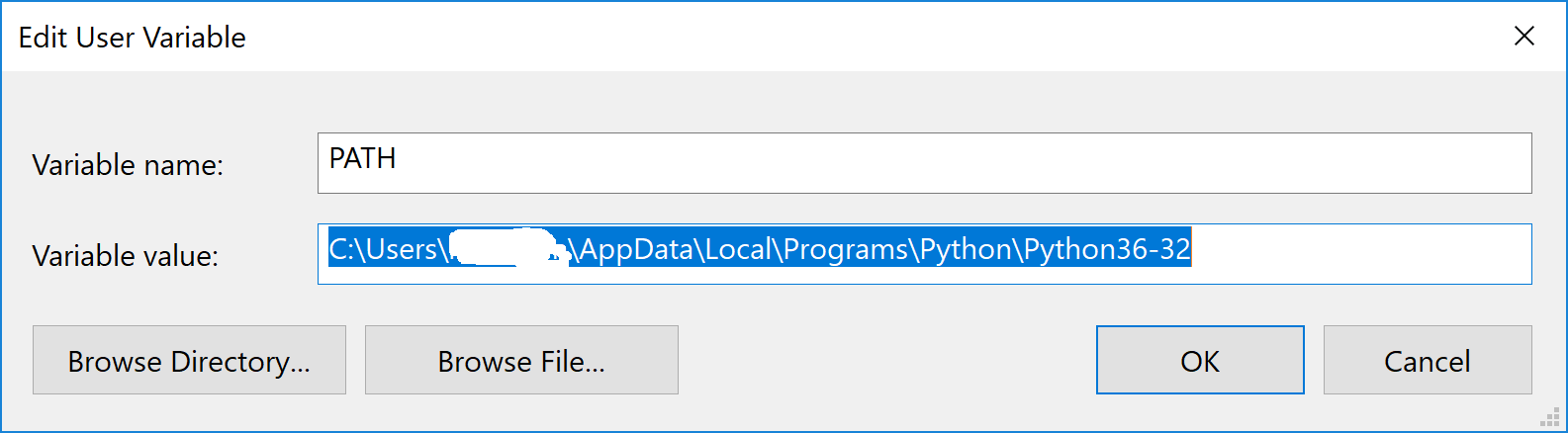
Go to

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

**Path :**

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

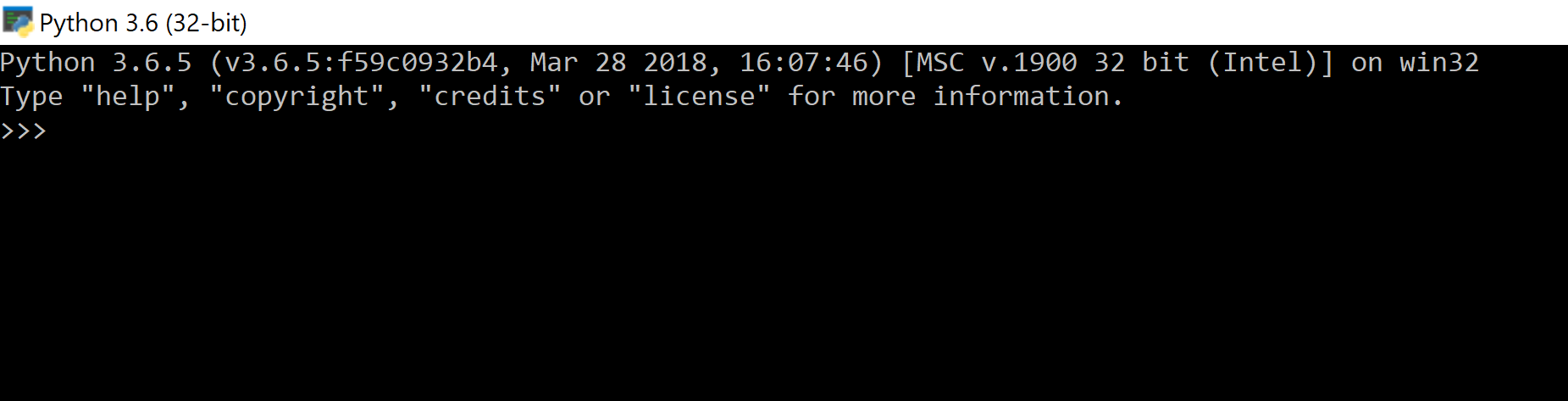




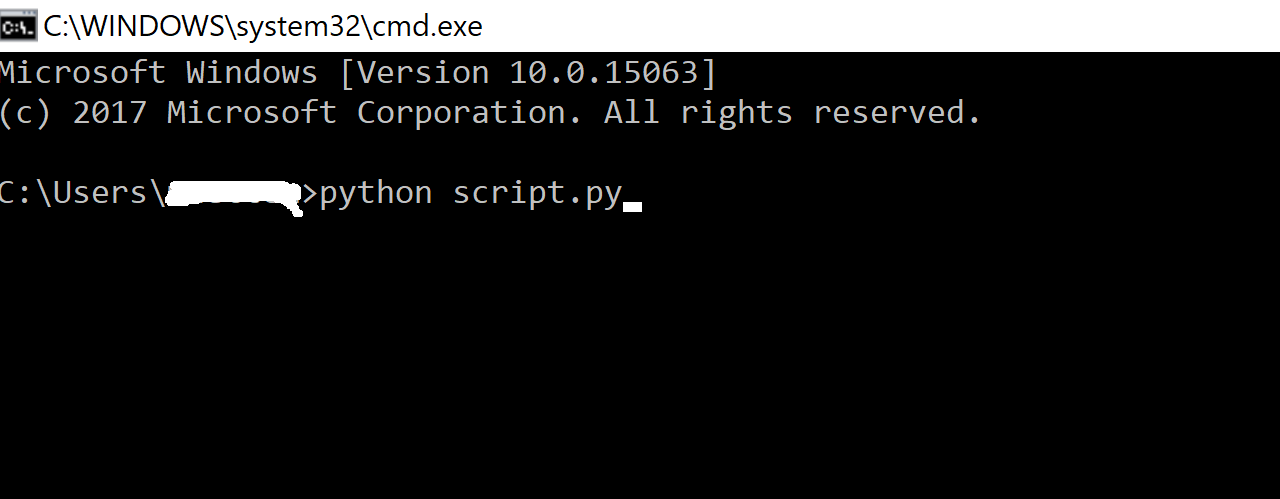
**1.5 Running Python:**

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.

For both Windows, Linux you can use different IDEs like Eclipse, **PyCharm**\*, Jupiter Notebook etc.,

**Advantages and Disadvantages of Python:**

Advantages:

1. Codereability, simple syntax
2. Dynamically typed
3. Improve the productivity
4. Interpreted lang
5. Free and open source
6. Vast support of library

Disadvantages:

**Slow speed:** Python is an **interpreted** language and **dynamically-typed** language. The line-by-line execution of code often leads to **slow execution**.

The dynamic nature of Python is also responsible for the **slow speed**of Python because it has to do the extra work while executing code. So, Python is not used for purposes where speed is an important aspect of the project.

**Not memory efficient:** To provide simplicity to the developer, Python has to do a little tradeoff. The Python programming language uses a **large amount of memory**. This can be a disadvantage while building applications when we prefer memory optimization.

**Weak in mobile computing:** Python is generally used in **server-side programming**. We don’t get to see Python on the client-side or mobile applications because of the following reasons. Python is **not memory efficient** and it has **slow processing power** as compared to other languages.

**Database Access:** Programming in Python is **easy** and **stress-free**. But when we are interacting with the database, it lacks behind.

The Python’s database access layer is primitive and underdeveloped in comparison to the popular technologies like **JDBC** and **ODBC**.

**Runtime error:** As we know Python is a dynamically typed language so the data type of a variable can change anytime. A variable containing integer number may hold a string in the future, which can lead to **Runtime Errors**.

**Interpreted vs Compiled time programming languages. Explain in detail:**

### **Compiled Languages**

Compiled languages are converted directly into machine code that the processor can execute. As a result, they tend to be faster and more efficient to execute than interpreted languages. They also give the developer more control over hardware aspects, like memory management and CPU usage.

Compiled languages need a “build” step – they need to be manually compiled first. You need to “rebuild” the program every time you need to make a change. In our hummus example, the entire translation is written before it gets to you. If the original author decides that he wants to use a different kind of olive oil, the entire recipe would need to be translated again and resent to you.

### **Interpreted Languages**

Interpreters run through a program line by line and execute each command. Here, if the author decides he wants to use a different kind of olive oil, he could scratch the old one out and add the new one. Your translator friend can then convey that change to you as it happens.

Interpreted languages were once significantly slower than compiled languages. But, with the development of [just-in-time compilation](https://guide.freecodecamp.org/computer-science/just-in-time-compilation), that gap is shrinking.

Examples of common interpreted languages are PHP, Ruby, Python, and JavaScript.

**.py vs .pyc files:**

.pyc files are generated automatically by the python interpreter when the python script is executed

.pyc contains the complied bycode of python source code

.pyc is not created for your main program file

**Explain Garbage Collection mechanism in detail**

With automatic memory management, programmers no longer needed to manage memory themselves. Rather, the runtime handled this for them.

There are a few different methods for automatic memory management. The popular ones use reference counting. With reference counting, the runtime keeps track of all of the references to an object. When an object has zero references to it, it’s unusable by the program code and can be deleted.

For programmers, automatic memory management adds a number of benefits. It’s faster to develop programs without thinking about low-level memory details. Further, it can help avoid costly memory leaks or dangerous dangling pointers.

However, automatic memory management comes at a cost. Your program will need to use additional memory and computation to track all of its references. What’s more, many programming languages with automatic memory management use a “stop-the-world” process for garbage collection where all execution stops while the garbage collector looks for and deletes objects to be collected.