# Python is a widely used High-level programming language.

# It is a general purpose interpreted,interactive,object-oriented 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).
* Python code is also available under GNU, GPL(General Public Licence).

PYTHON OVERVIEW :

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

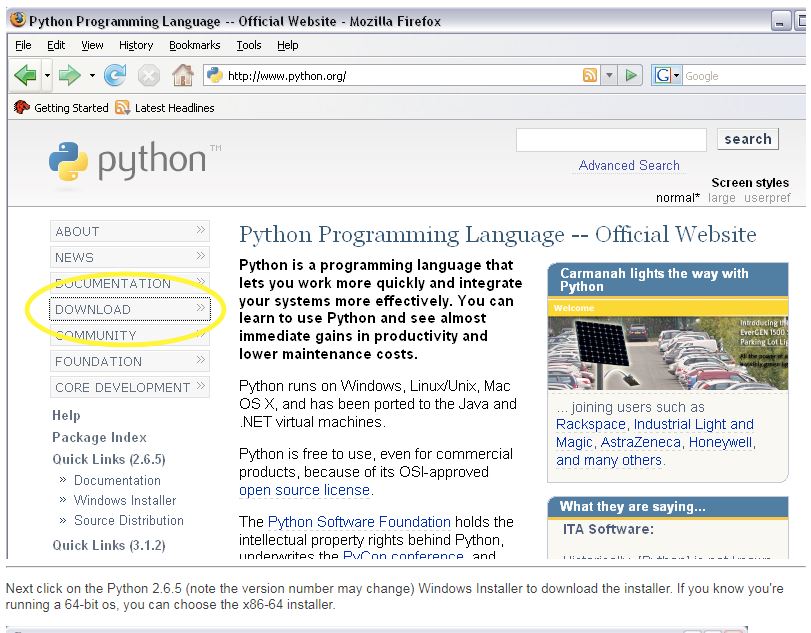
* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your 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

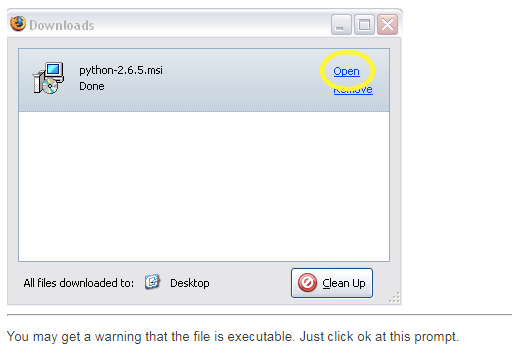
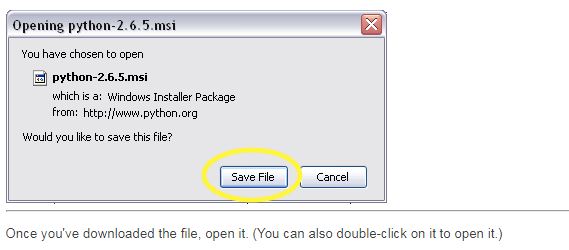
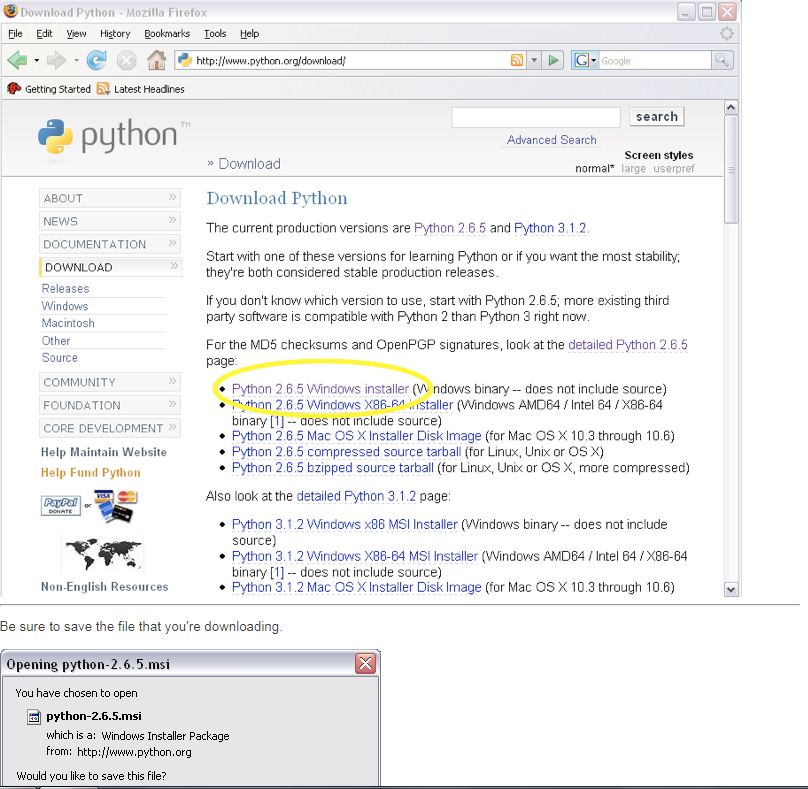
PYTHON ENVIRONMENT SETUP :

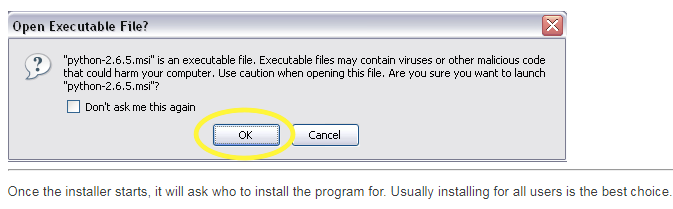
Python is available on a wide variety of platforms including Linux and Mac OS X.Here are the steps to install Python :

1.Open a Web browser and go to [https://www.python.org/downloads/.](https://www.python.org/downloads/)

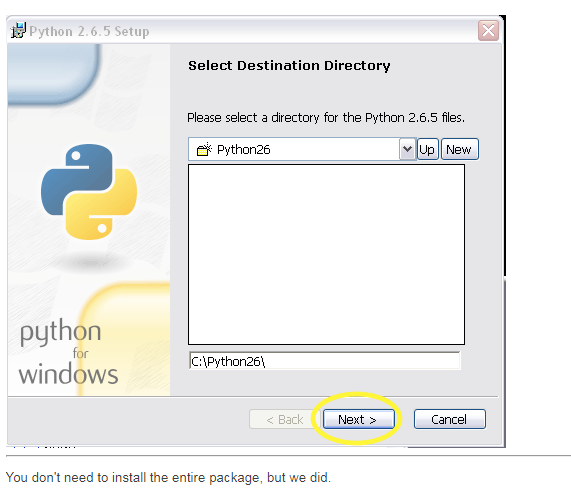
Follow the link for the Windows installer *python-XYZ.msi*file where XYZ is the version you need to install.

1. To use this installer *python-XYZ.msi*, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
2. Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

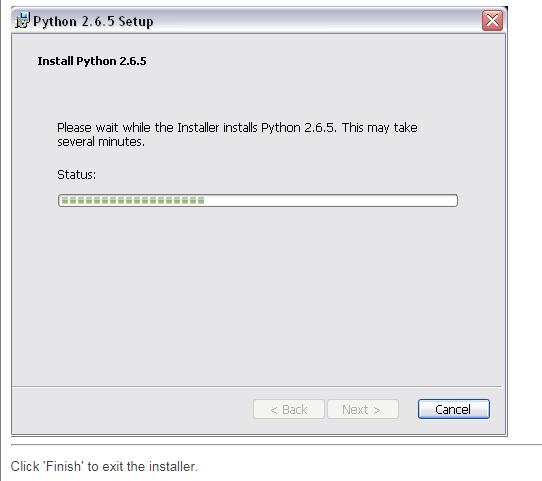


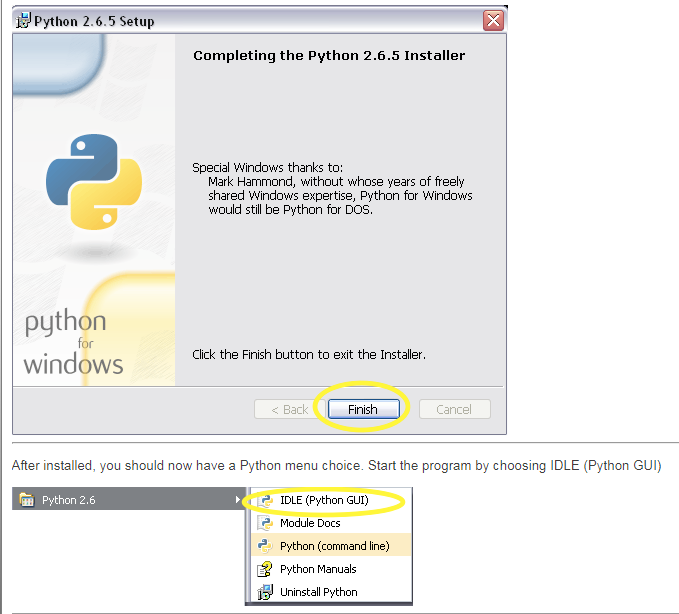


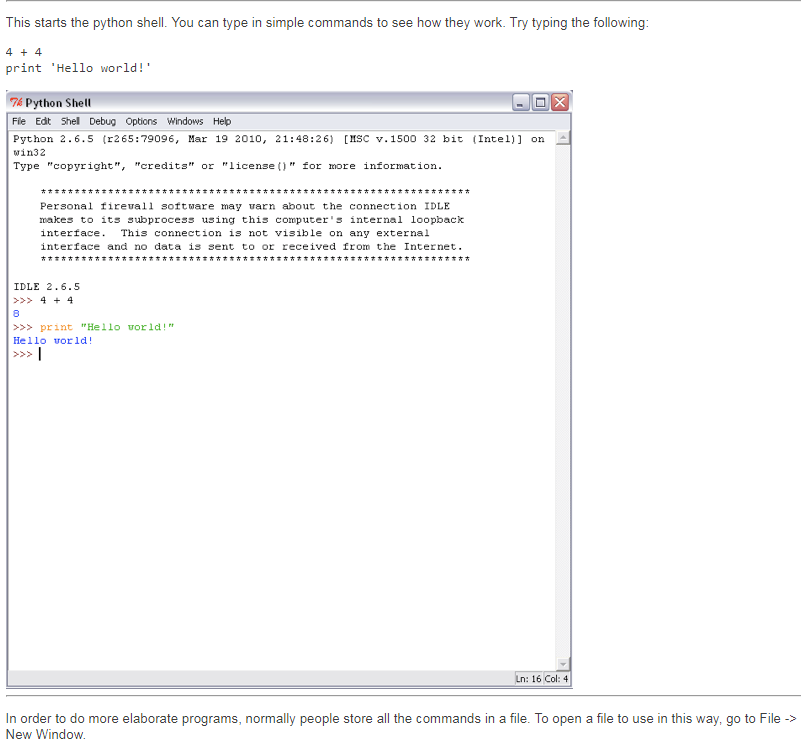


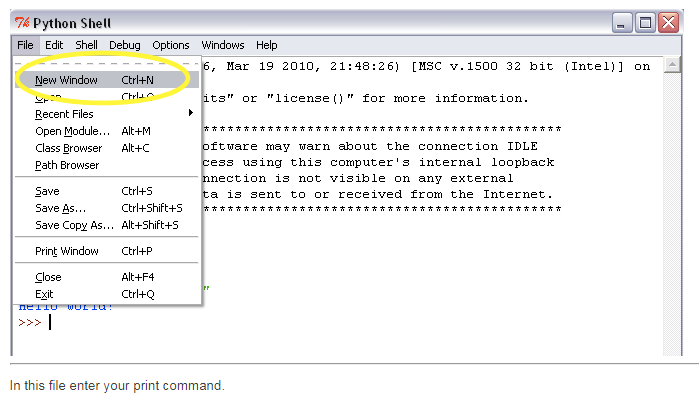


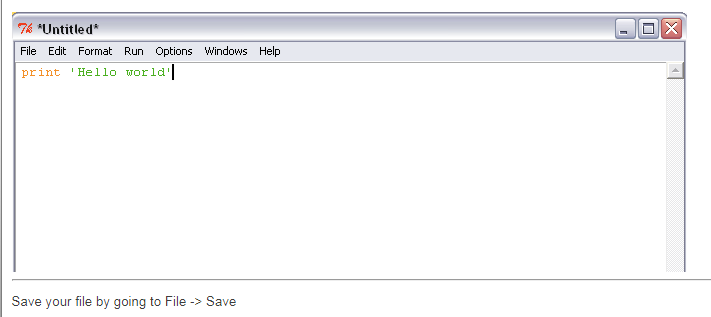




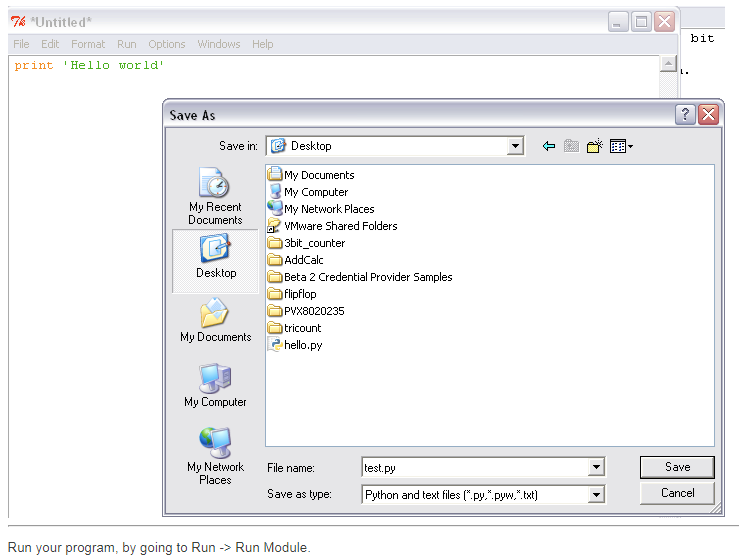


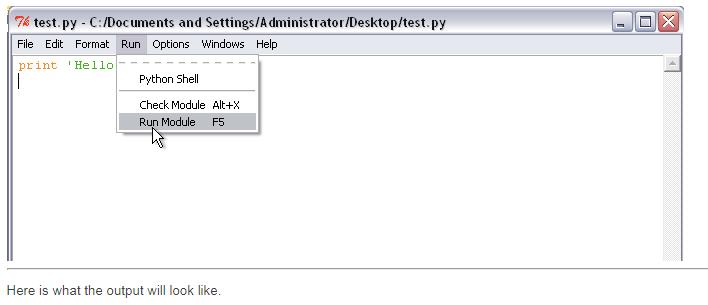


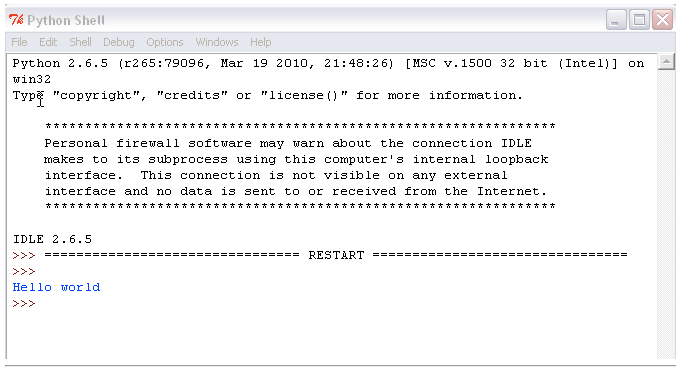








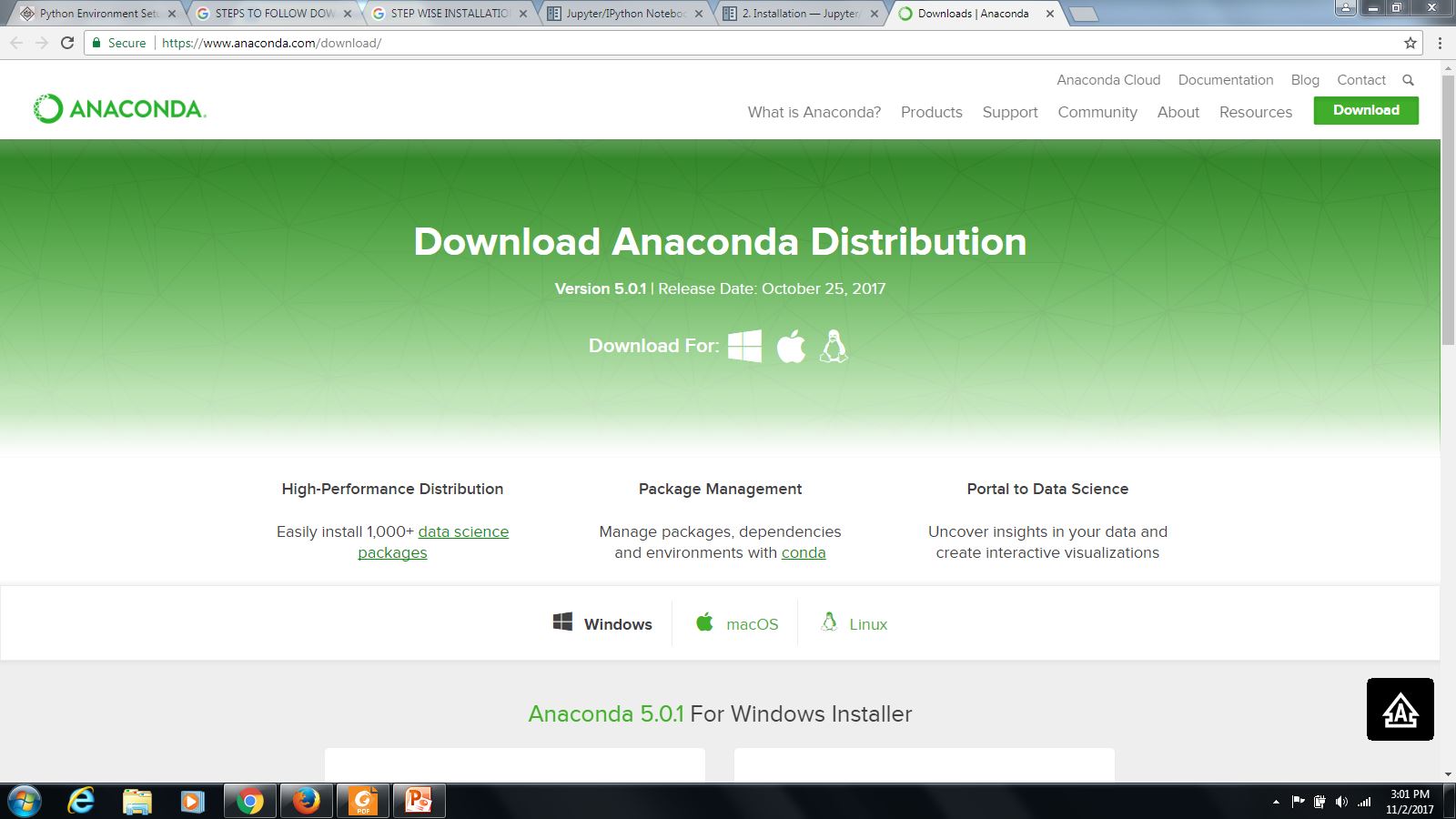


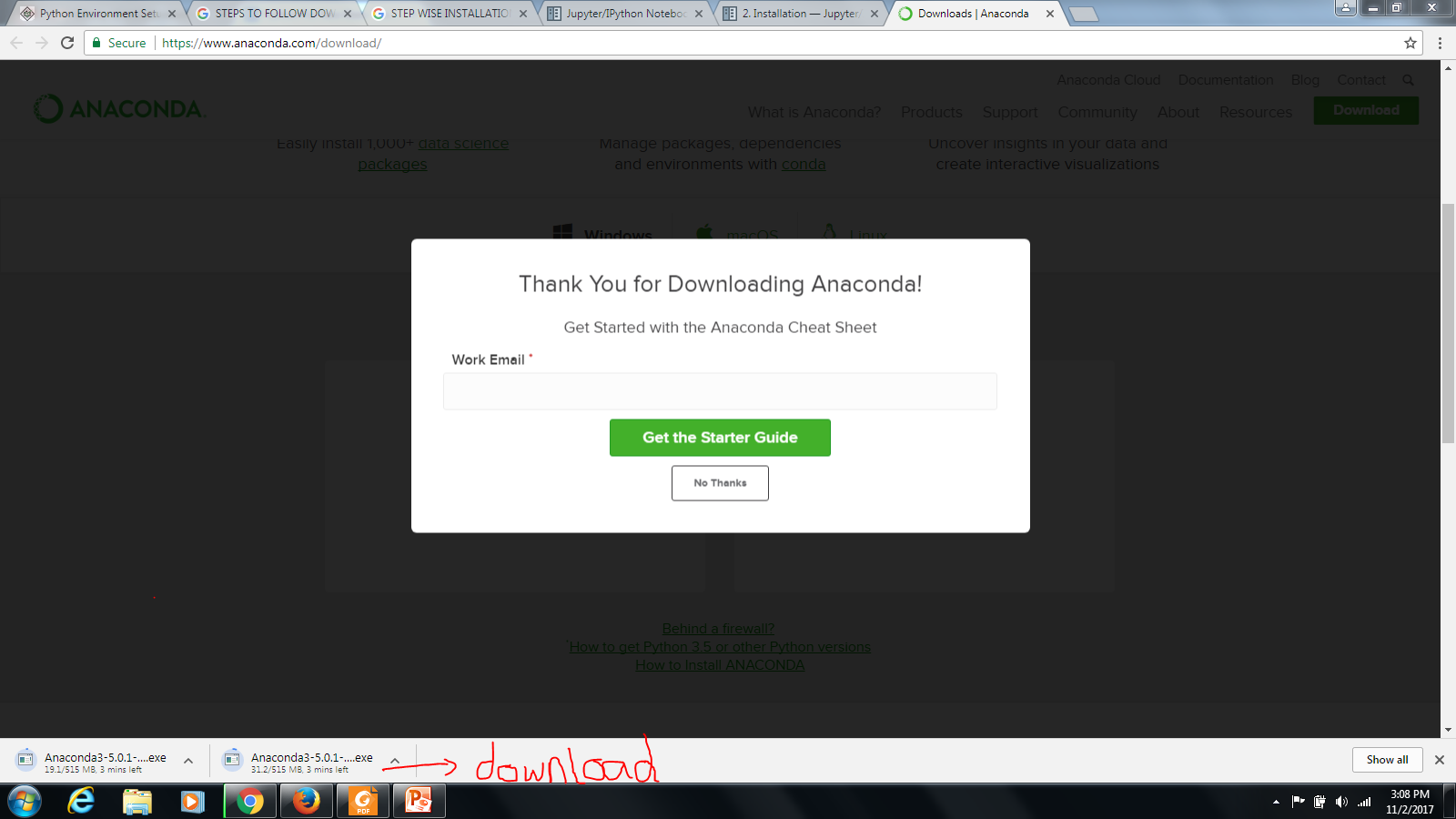
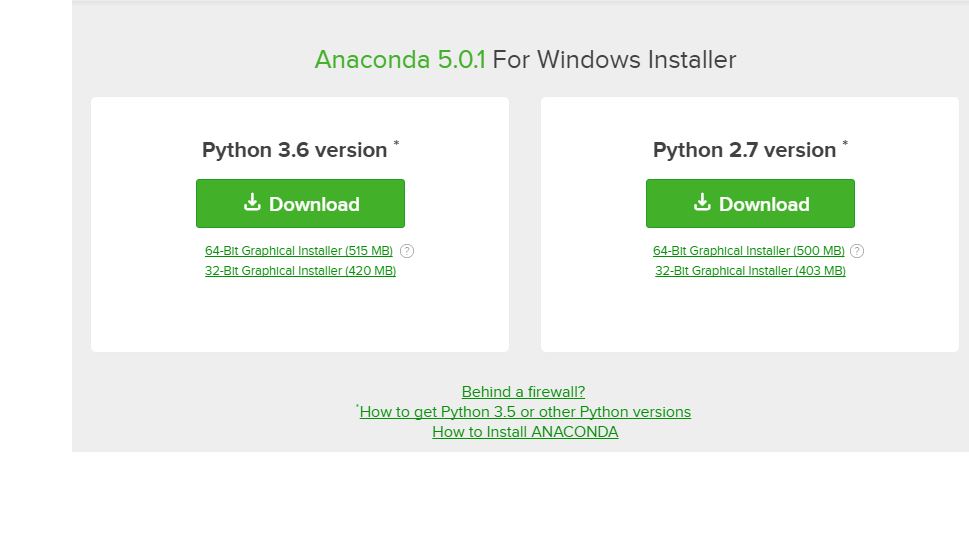


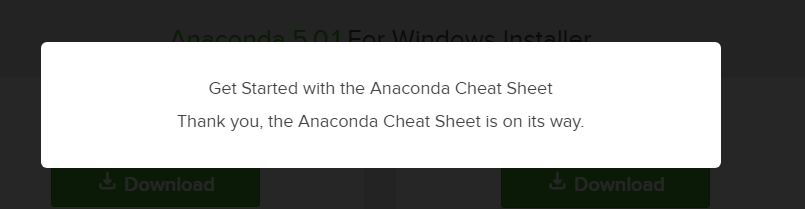
This is one of the method to write your python script and execute.Another platform to run your python code is Jupyter Notebook.The **Jupyter Notebook** is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.Here is the detail information how to install and use this Jupyter Notebook

* While Jupyter runs code in many programming languages, **Python** is a requirement (Python 3.3 or greater, or Python 2.7) for installing the Jupyter Notebook.
* We recommend using the [Anaconda](https://www.continuum.io/downloads) distribution to install Python and Jupyter. We’ll go through its installation in the next section.

INSTALLING JUPYTER USING ANACONDA “

1. Anaconda conveniently installs Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science
2. Here the following are the steps for downloading ANACONDA.
3. Download [Anaconda](https://www.continuum.io/downloads). Downloading Anaconda’s latest Python 3 version (currently Python 3.6.2).
4. Install the version of Anaconda which you downloaded, following the instructions on the download page.





Now your application is download is successfully and you have installed Jupyter Notebook.

Alternative for experienced Python users: Installing Jupyter with pip :

**NOTE : Jupyter installation requires Python 3.3 or greater, or Python 2.7. IPython 1.x, which included the parts that later became Jupyter, was the last version to support Python 3.2 and 2.6**

As an existing Python user, you may wish to install Jupyter using Python’s package manager, pip, instead of Anaconda.

First, ensure that you have the latest pip; older versions may have trouble with some dependencies:



Then install the Jupyter Notebook using:



Now JUPYTER NOTEBOOK is installed

BASIC STEPS:

Start JUPYTER NOTEBOOK server from the command line ;

* Easy-to-Read
* Easy-to-Maintain
* Portable
* Scalable
* Extendable
* Databases
* Broad Standard Library
* Interactive Mode
* GUI Programming
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows,
* **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

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.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, CORBA, and Java.

BASIC SYNTAX OF PYTHON :

Python language has many similarities to Perl,C & JAVA. But there are some differences between the languages.There are basically two modes (types) of programming.They are :

* Interactive Mode Programming
* Script Mode Programming

INTERACTIVE MODE PROGRAMMING :

Invoking the interpreter without passing a script files as a parameter brings up. In this the program was scripted in the python shell using IDLE(Python 3.6.2) as follows :

**Program** :

**#!/usr/bin/python3**

print (“Hello Python!”)

STEP 1 :

Script the program in the notepad



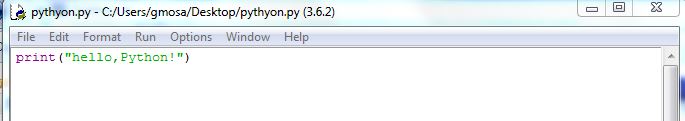
STEP 2 :

Then save the file now

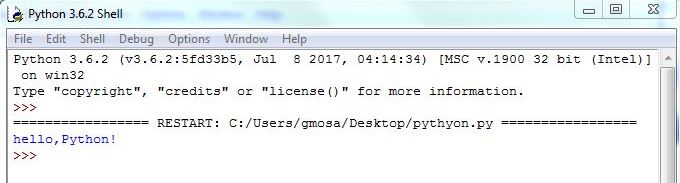


Right click this file go to Edit with IDLE and then,edit with IDLE 3.6(32 bit).

STEP 3 :



Click RUN to run this module.

STEP 4 :

This is the output in Interactive mode scripting.

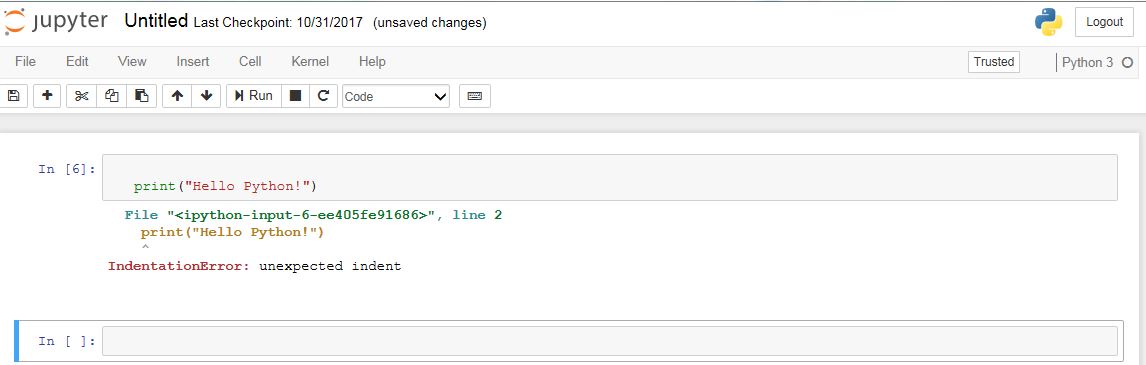
SCRIPT MODE PROGRAMMING :

Invoking the interpreter with a script parameter begins execution of scripts & continuous until the script is finished. When the script is finished,the interpreter is no longer active.

* .py Extension for python files
* Test.py : xxxx for following source code

Example :

print (“Hello Python!”)

This is the error message thrown when we didn’t mention the header [**#!/usr/bin/python3]**

Output :****

PYTHON IDENTIFIERS :

Identifier is a name used to identify a variable, function, class, module or other object.

**RULES :**

1. An identifier starts with a letter A to Z or a to z or an underscore (\_) followed by zero or more letters, underscores and digits (0 to 9).
2. Does not allow punctuation characters such as @, $, and % within identifiers.
3. It is a case sensitive programming language**.**
4. Class names start with an uppercase. Remaining identifiers start with a lowercase.
5. An identifier with a single underscore indicates that the identifier is private.
6. An identifier with two underscores indicates a strong private identifier.
7. If the identifier ends with two trailing underscores, the identifier is a language-defined special name.

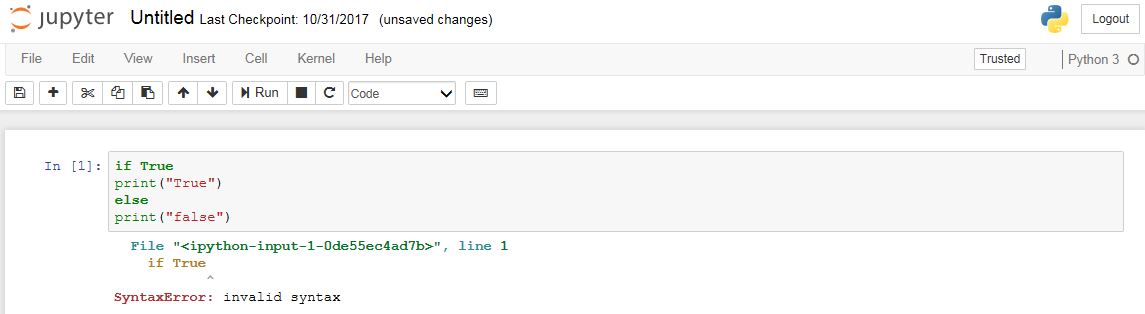
RESERVED WORDS :

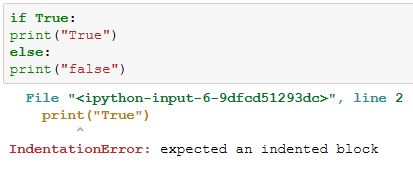
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Try | and | If | import | is |
| def | assert | except | lambda | in |
| del | class | for | print | or |
| elif | continue | from | pass | not |
| break | exec | finally | return | with |
| else | as | global | raise | yield   |  | | --- | | while | |

LINES & INDENTATION :

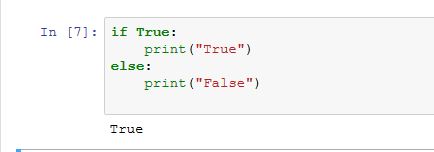
In Python lines and indentations should maintain the sequence according to one to other. The following examples gives you brief about the lines and indentation in Python.

Example :



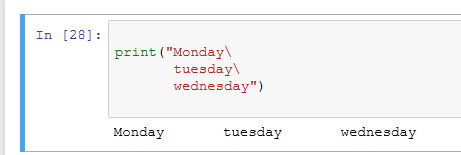


If the program throws an **IndentationError** then press “Delete & Enter” the final output is as follows :



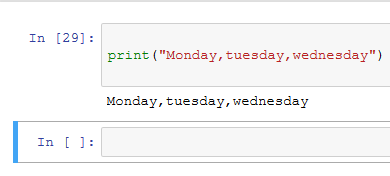
MULTI-LINE STATEMENTS :

Example :



Statements contained within the [], {}, or () brackets do not need to use the line continuation character.

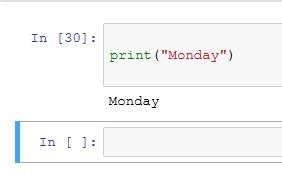
Example :

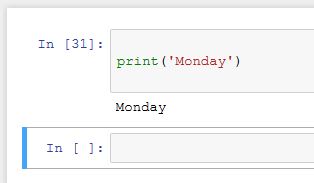


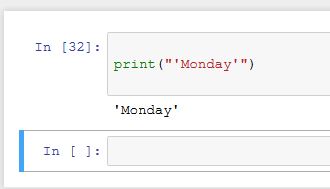
QUOTATIONS IN PYTHON :

Python accepts these quotations (“),(‘),(“ ‘\_ ‘ “).

Example :





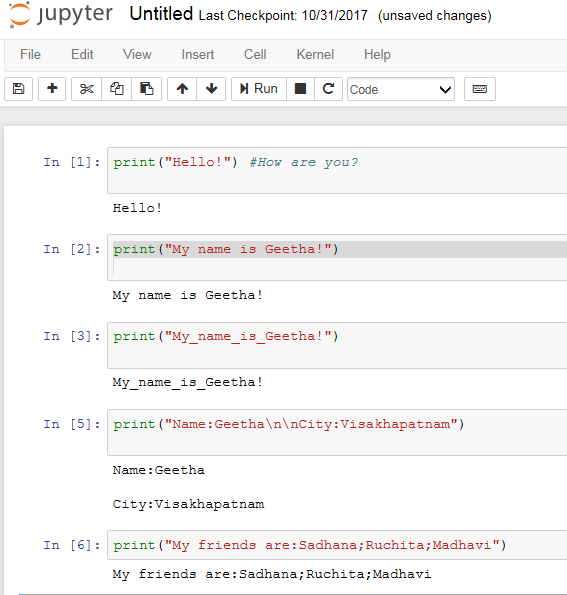


COMMENTS IN PYTHON :

* “ # ” next to this return comments are not displayed.
* Python ignores blank lines (Containing whitespaces,possibly with comment).
* “ \n\n “ creates two new lines before displaying the actual line.
* Semicolon (**;**) allows multiple statements on a single line.

Examples :

Above mentioned rules are executed. Here are some examples below:

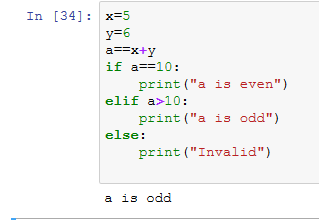


MULTIPLE STATEMENT GROUP AS SUITS :

Group of individual statements,which makes as a single code suites in python. Compound or complex statements such as **if,while,def & class** require a header line & a suit.

* Header line begins with statement & ends with colon(**:**).

Example:



VARIABLE TYPES :

Variables are nothing but reserved memory locations to store values. It means, when you create a variable, you reserve some space in the memory by assigning different data types to the variables.You can store integers, decimals or characters in these variables.Basically python has five data types.they are :

* Numbers
* String
* List
* Tuple
* Dictionary

Python supports four different numerical types.They are :

* Integer(int) --->signed integers
* Float ---> floating point real values
* Complex ---> complex numbers
* Long --->long integers,they can also represent in octal & hexadecimal

|  |  |  |
| --- | --- | --- |
| **int** | **float** | **complex** |
| 10 | 0.0 | 3.14j |
| 100 | 85.50 | 45.j |
| -500 | -61.6 | 9.322e-36j |
| 020 | 21.9+e34 | .876j |
| -0234 | -56. | -.6545+0J |
| -0x234 | -21.9e34 | 3e+26J |
| 0x435 | 70.2-E12 | 3.2e-26j |

**NOTE :**

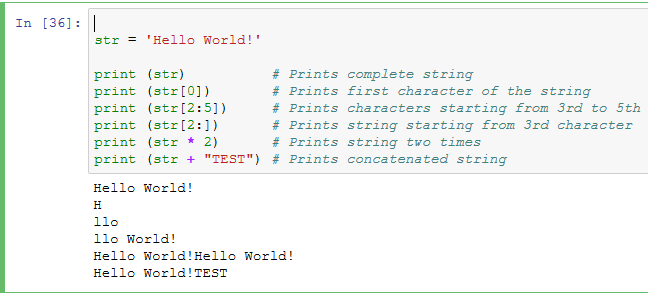
* L:Uppercase
* J,E,e,:Real numbers (x+y)
* j:Imaginary unit (x+yj)

STRINGS :

Strings in python are defined as a continuous set of characters represented in quotation mark. Both single and double quotes are allowed.

* Subsets can be taken using slice operator **([ ] & [:]).**
* Indexes begins at **‘0’** and way from **‘-1’** at the end.
* **“+”** indicates string concatenation.
* **‘\*’** indicates repetition operator.

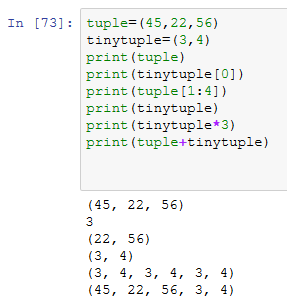
Examples:



TUPLES :

A tuple is another sequence data type that is similar to the list.

Example :

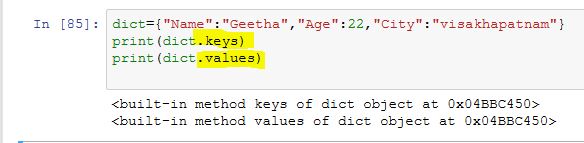


DICTIONARIES :

Dictionaries are enclosed by curly braces (**{ }**) & value can assigned and accessed by square braces (**[ ]**).

Example :





DATA TYPE CONVERSION :

To perform conversions between the built-in types. To convert between types, you simply use the type-names as a function.We have some function with descriptions mentioned below:

1. **int(x [,base]) :** Converts x to an integer.The base specifies the base if x is a string.This follows the logarithm rules.
2. **float(x) :** Converts x to a floating-point number.
3. **complex(real [,imag]) :** Creates a complex number.
4. **str(x) :** Converts object x to a string representation.
5. **repr(x) :** Converts object x to an expression string.
6. **eval(str) :** Evaluates a string and returns an object.
7. **tuple(s) :** Converts s to a tuple.
8. **list(s) :** Converts **s** to a list.
9. **set(s) :** Converts **s** to a set.
10. **dict(d):** Creates a dictionary. **d** must be a sequence of (key,value) tuples.
11. **frozenset(s) :** Converts **s** to a frozen set.
12. **chr(x) :** Converts an integer to a character.
13. **unichr(x) :** Converts an integer to a Unicode character.
14. **ord(x):**Converts a single character to its integer value.
15. **hex(x):**Converts an integer to a hexadecimal string.
16. **oct(x):** Converts an integer to an octal string.

BASIC OPERATORS :

Python supports seven types of operations.They are :

* Arithmetic Operators
* Relational Operators
* Assignment Operators
* Logical Operators
* Bitwise Operators
* Membership Operators
* Identity Operators

ARITHMETIC OPERATORS :

In this we have seven mathematical operators,which basically we use.They are :

* Addition : ‘**+**’
* Substraction : ‘**-**’
* Multiplication: ‘**\***’
* Division : “**/**”
* Modulus: “**%**”
* Exponent : “ **\*\*** “
* Floor Division : “**//**”

**NOTE:**

The division of operands where the result is quotient in which the digits after the decimal point are removed. But if one of the operands are **-**ve,the result is floored rounded away from zero (towards **-**ve ,infinity).

Examples:

