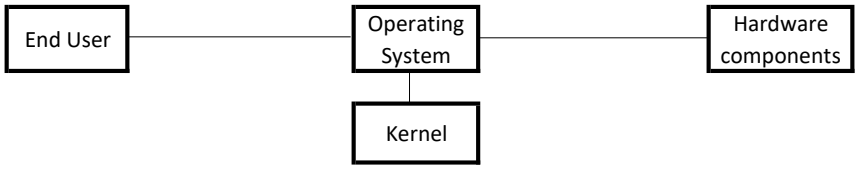
PYTHON BASICS

Python is formally an interpreted language. Commands are executed through apiece of software known as the Python interpreter.
Integrated development environments (IDEs) provide richer software development platforms for Python,

What is kernel ?
The kernel is a computer program at the core of a computer's operating system and generally has complete control over everything in the system. It is the portion of the operating system code that is always resident in memory and facilitates interactions between hardware and software components.



Identifiers, Objects, and the Assignment Statement

During assignment statement, such as **temperature = 98.6**

temperature as an **identifier**
98.6 as **object**

The identifier temperature references an instance of the float class having value 98.6.

Note:
An i**dentifier** cannot begin with a numeral (thus 9lives is an illegal name), and that there are 33 specially reserved words that cannot be used as identifiers

An i**dentifier** can be associated with any type of object, and it can later be reassigned to another object of the same (or different) type.

temperature = temperature + 5.0
The execution of this command begins with the evaluation of the expression on the **right-hand side** of the = operator. That expression, temperature + 5.0, is evaluated based on the existing binding of the name temperature, and so the result has value 103.6, that is, 98.6 + 5.0. That result is stored as a new floating-point instance, and only then is the name on **the left-hand side** of the assignment statement, temperature, (re)assigned to the result.

```
In:     temp = 20
        original = temp
        temp = temp + 1
Ou:     temp
                21
        original
                20
```
The temperature identifier has been assigned to a new value, while original continues to refer to the previously existing value.

Creating and Using Objects

Instantiating an object is to invoke the constructor of a class.
For example, if there were a class named Widget, we could create **an instance of that class using a syntax such as w = Widget(),**

Calling Methods

data.sort( )
The expression to the left of the dot identifies the object upon which the method is invoked. Often, this will be an identifier (e.g., data)
We can use the dot operator to invoke a method upon the immediate result of some other operation.
Eg. The syntax response.lower( ).startswith(y) first evaluates the method call, response.lower( ), which itself returns a new string instance, and then the startswith(y) method is called on that intermediate string.

Python's Built-In Classes

A class is **immutable** if each object of that class has a fixed value upon instantiation that cannot subsequently be changed

Mutable	list	set	dict			
Immutable	bool	int	float	tuple	str	frozenset

All of the classes support a traditional constructor form that creates instances that are based upon one or more existing values. Iterable types include all of the standard container types (e.g., strings, list, tuples, sets, dictionaries).

Logical Operators		**Equality Operators**		**Comparison Operators**	
not	unary negation	**is**	same identity	**<**	less than
and	conditional and	**is not**	different identity	**<= l**	ess than or equal to
or	conditional or	**==**	equivalent	**>**	greater than
		!=	not equivalent	**>=**	greater than or equal to

Arithmetic Operators		**Bitwise Operators**	
=+	addition		
–	subtraction	**~**	bitwise complement
*****	multiplication	**&**	bitwise and
/	true division	**|**	bitwise or
//	integer division	**ˆ**	bitwise exclusive-or
%	the modulo operator	**<<**	shift bits left, filling in with zeros
		>>	shift bits right, filling in with sign bit

If variables **n and m** represent respectively the **dividend and divisor** of a quotient nm , and that **q = n // m** and **r = n % m**. Python guarantees that **q*m + r** will equal **n**.

Open Jupyter Notebook

Step 1 Open anaconda prompt and type "jupyter notebook"
CMD = Command Prompt (which is not required.)
Step 2 Under "New tab" on top right corner select "python 3 (pykernel)"
You will get a new jupyter notebook screen
Jupyter is just for begineers for practice purpose and serves less purpose when using professionally.

Features of Jupyter notebook

File tab	New notebook	Open the new jupyter file You can change the name of the file directly by clicking the title on the top of the screen
	Open	Open the exixting file
	Download as	When you want to upload your assignments on github, download in "Notebook (.ipynb)" format. Select the location where you want to download and upload the same file on github.
Ribbon	Add a new cell Remove the existing cell Copy a cell Move between cells Enter+shift to run the command Stop the run command Restart your kernel	

Jupyter notebook is only for working backend to write code. Actual execution of code is done in python compiler.

Sometimes due to variable overwrite issue, results will not be updated and will show star markes to the left.
It means that jupyter notebook is unable to connect to python compiler.
Solution:
Under kernel tab, click on "interrupt" which will interrupt your kernel.
or
Restart your kernel

or
Restart and clear output
or
Restart and run all the code you have written
or
Reconnect your kernel
or
Shutdown your kernel.
or
You can change your kernel in case you have multiple versions of python or multiple parallel kernels installed.

Incase you want to write anything other than code

Step 1 Type or copy paste the text in any cell.

Step 2 Change the format in ribbon tab from "code" to "Markdown"/'Raw NBconvert"
"Markdown" will give you things without a box
"Raw Nbconvert" will give a box to the content

If there is a data in the form of excel sheet and you want to filter out data you want:

Step 1 Read the excel sheet

Step 2 Store the file as a variable

Step 3 Perform required operations on the variable.

Note: When using 's in a string variable try to use double quotes for string variable or else ull get an error

Note:

```
a = 10
b = 12
c = 13
```

or

```
a , b , c = 10 , 12 , 13
```

or

```
a,b,c = (10,12,13)
```