

# Python Fundamentals

## day 2

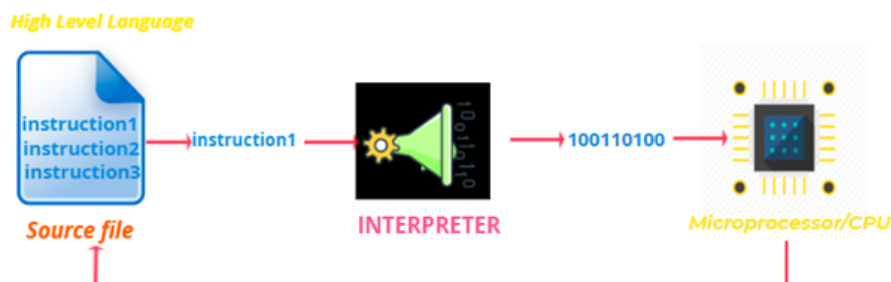
### Today's Agenda

- Programming in Python
- Interactive mode & Script mode
- Difference b/w interactive & script mode
- Memory allocation
- Object Oriented Programming
- Principles of OOP



### Programming in Python

After knowing what is Python and different features of it and the history behind. Let us now get to know if python is an interpreted programming language or if it is compiled programming language



This is how a python file executes. Each instruction is given to the interpreter which converts it into machine level and feeds to CPU to get the output. And again to get the next instruction we have to go back to the source file and continue the process. So no doubt python is an interpreted language.

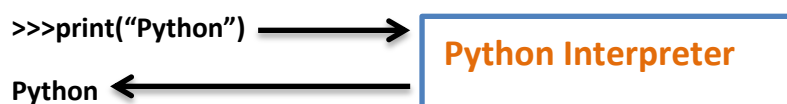
# Interactive mode & Script mode

Let us now look into different modes of writing a code in python:

## Interactive mode

The interactive mode involves running your codes directly on the Python shell which can be accessed from the command line/terminal of the operating system. Single line fed to the interpreter is executed at that moment of time, and waits for next line to be entered. Each line is given to the interpreter which internally converts HLL to MLL and then gives the respective output.

```
>>>print("Python")
Python
```



**Python Interpreter**

```
>>>a = 10
>>>b = 20
>>>c = a+b
>>>print( c )
30
```

// similarly all other lines, one by one are given to interpreter.

// lines starting with >>> Represents the code or the input to the interpreter. Lines without >>> are the outputs

And whenever a variable with value is fed to the interpreter a container gets created internally with the value in it. As shown below:

```
c 30
b 20
a 10
```

And to exit this interactive mode you should give `quit()` as input.

## Pros and cons of Interactive Mode

### Pros:

- It is great for single line or smaller codes.
- Interactive mode is a good way to play around and try variations on syntax.

### Cons:

- Once you come out of interactive mode, you cannot revisit the previous code.
- It is harder to edit longer programs or even existing program.
- Codes cannot be saved in interactive mode.



## Script Mode

To overcome the drawbacks of interactive mode we have the second mode of writing a code called as Script mode. Let us know different platforms of typing a python code in script mode.

- Notepad **//basic text editor**
- Microsoft Word **//basic text editor**
- Different IDE (**IDLE, Spider, PyCharm, Jupyter Notebook**)

Here let us get to know how to run a python script in command line.

### Script mode in command line

We know that if we type python and press enter we are into interactive mode like shown below:

```
C:\Users\rooman>python
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> //entered the interactive mode
```

Let us type the same addition code in python script and see how it works.

```
print("Python")
a = 10
b = 20
c = a+b
print(c)
```

Let us see how to run a python script in command line.

Syntax:

```
C:\Users\rooman>python <python script name>.py
```

Note: Before running this command, make sure you are in the same directory as of the python script you want to run.

Output:

```
C:\Users\rooman>cd OneDrive
```

```
C:\Users\rooman\OneDrive>cd Desktop
```

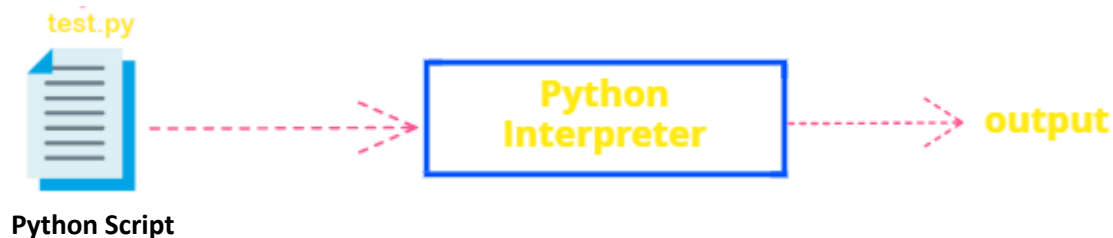
```
C:\Users\rooman\OneDrive\Desktop>cd python
```

```
C:\Users\rooman\OneDrive\Desktop\python>python test.py
Python
30
```

In the script mode, you have to create a file, give it a name with a **.py** the extension then run your code. The file which contains python code and has to be executed is called as **python script**. The

script mode is recommended when you need to create large applications.

As we have the python script in HLL, conversion must happen as below:



### Script mode + Interactive mode

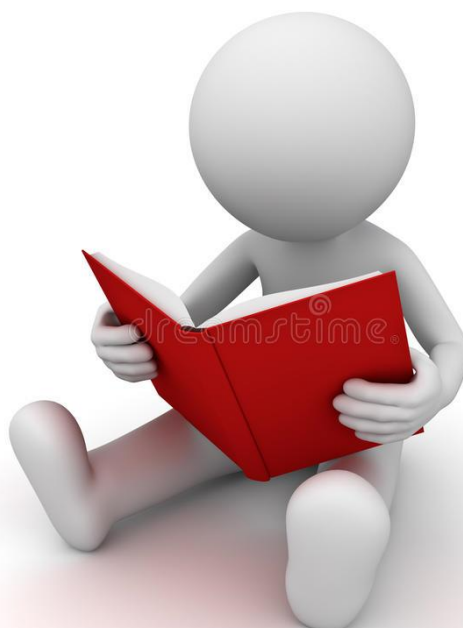
Let us see how to enter interactive mode after running the python script

Syntax:

```
C:\Users\rooman\OneDrive\Desktop\python>python -i <python script>.py
```

Command line execution:

```
C:\Users\rooman\OneDrive\Desktop\python>python -i test.py
Python
30
>>> d = a*b
>>> print(d)
200
```

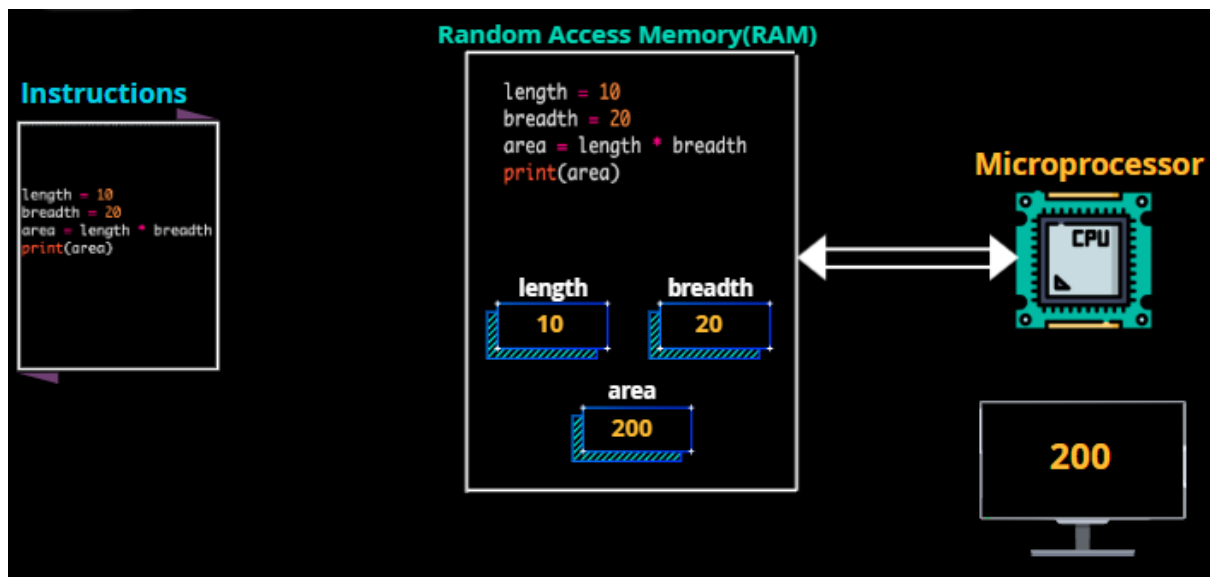


## Difference b/w interactive & script mode in python

Interactive mode	Script Mode
A way of using the python interpreter by typing commands & expressions at any prompt.	A way of using the python interpreter to read & execute statements in a script
Can't save and edit the code.	Can save and edit the code
If we want to experiment with the code, we can use interactive mode.	If we are clear about the code, use script mode.
We cannot save the statements for further use and we have to retype.	We can save the statements for further use and no need to retype all the statements to return them.

## Memory allocation

On one end we have the instructions and on the other end microprocessor. To give these instructions as the input to microprocessor we need to store it so that the microprocessor has direct access to the instructions. To store these instructions we have something called as **RAM (Random Access Memory)**. RAM is a temporary memory directly connected to the microprocessor, and the main purpose of RAM is to store the instructions before it is given to microprocessor. And then line by line instructions gets converted to MLL and then given to microprocessor.



RAM is the most important memory in the computer where all execution of the program happens on the RAM. In fact if the instructions are on the RAM then only execution can happen if not execution cannot be performed.

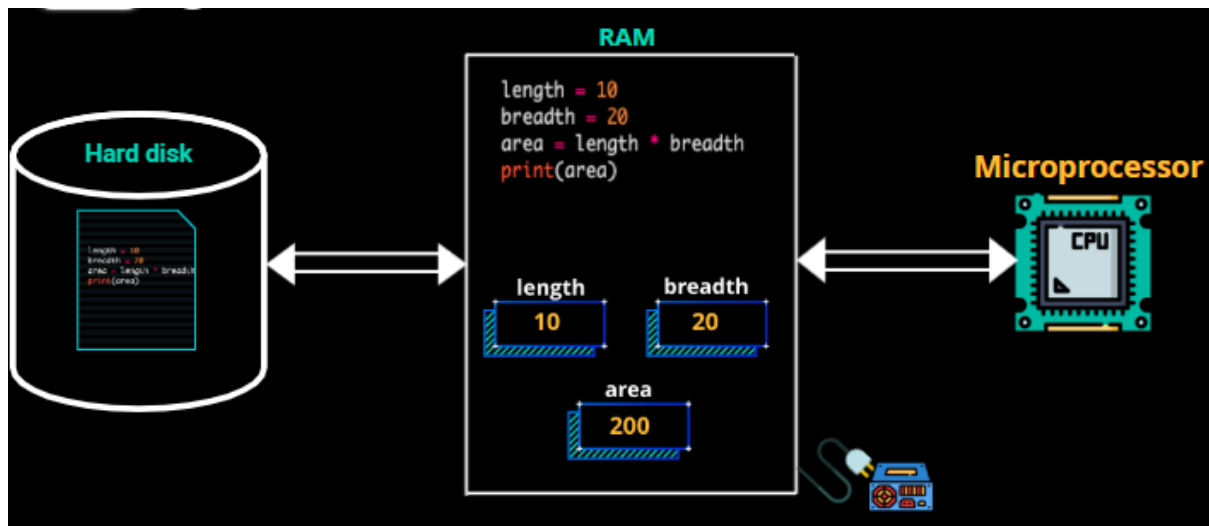
RAM is a semiconductor technology device just like the microprocessor, which means it is made of transistors. So therefore RAM always needs constant supply of electricity for it to work. Once the supply is gone all the instructions and operations happening inside the RAM vanishes. That is the reason RAM is said to be volatile in nature.



### Hard Disk



RAM is the main memory, as it is volatile in nature we have secondary memory which is called as Hard disk which is a magnetic device. To use the instructions again and again you need to make a copy of those instructions and store it in hard disk. Those instructions while are store in hard disk are called as file.



The process of taking a copy from RAM and storing in hard disk is called as **saving**. And **loading** is a task where we are taking file and transferring it to RAM.

All computers have these two memory devices namely hard disk and RAM. Where hard disk is used for permanent storage and RAM is used for temporary storage.

**Note:** Microprocessor is connected only to RAM. And RAM is connected to the hard disk.

## Object Oriented Programming

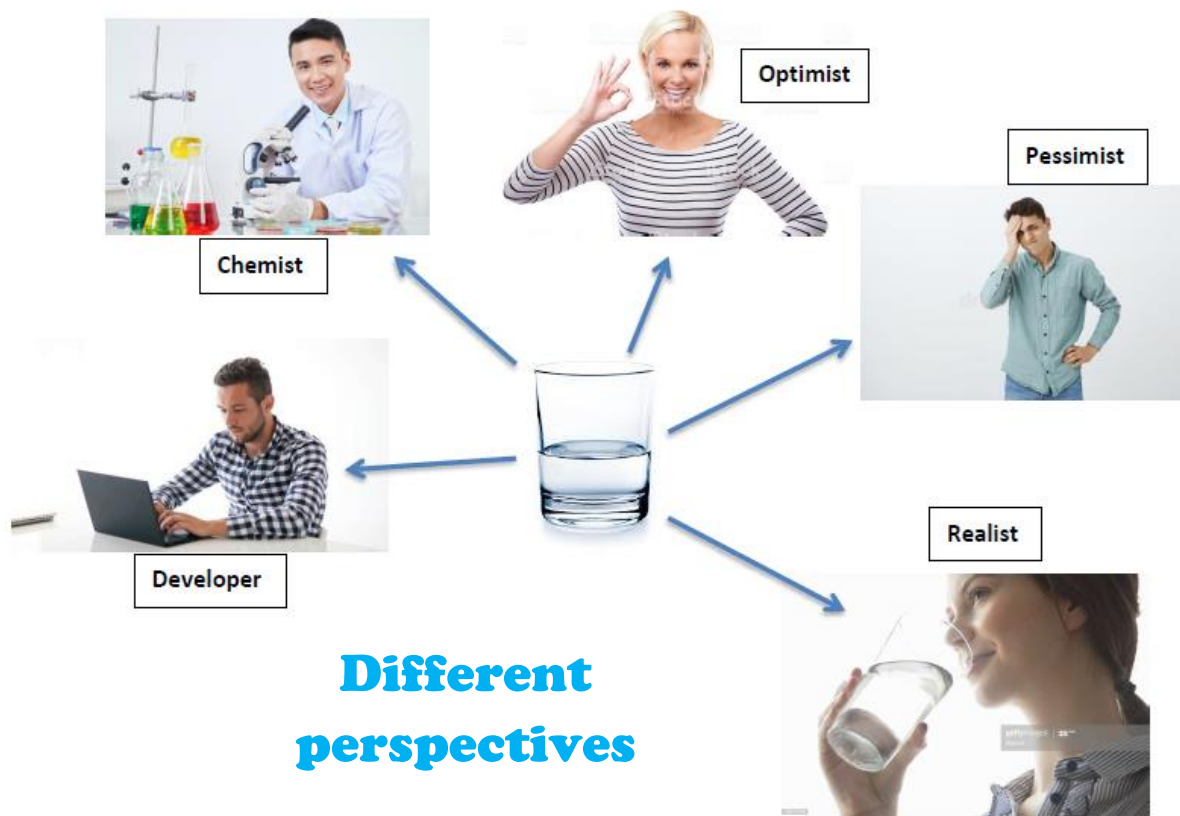
We have two different types of programming languages, which are structural programming language and the second is object oriented programming (OOP) language. Here we will be focusing on OOP language.

Before going ahead with the **object orientation**, let us get to know **what is orientation?**

**Orientation → Perspective → Way of looking at something**

Let us take an example of glass of water and try to understand in better way:





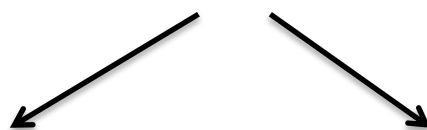
A chemist will see the glass of water as liquid + gas, an optimistic person will look at it as half full glass, a pessimistic person will see it as half empty glass, a realist will see it as glass of water, but a **developer will always view it as an object.**

Object Orientation is the way of looking at this world as a collection of objects. In this world no object is completely useless. All objects are in constant interaction with each other. No object exists in isolation.

# Principles of OOP

- View everything around you as objects.
- Every object belongs to a type. Where type does not exist but the objects of the type exist in reality.
- Every object has two parts:
  1. State of an object /properties
  2. Behaviour of an object/actions
- To handle the properties of the states of an object we have a concept called as **data types**
- Similarly to handle behaviour of an object we have a concept called as **functions**

Let's see the same with example of bike



## State/properties:

- Brand
- Price
- Cylinder capacity (cc)
- Colour

## Behaviours:

- start( )
- change\_gear( )
- accelerate( )
- brake( )