# **Introduction**:

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions- Python 2 and Python 3. Both are quite different.

1. Scripting language(interrupted) features from Perl & Shell script.
2. Object oriented concept from C++.
3. Function programming from C, without having Class you can call function.
4. Syntax borrow from C & ABC language.
5. Free Ware and Open Source.
6. High level programming language(Machine level) memory and space management by internally.
7. Platform Independent -> write once and run anywhere.
8. Portability->Migrating easily->python program any machine.
9. Dynamically Typed->no need to declare variable type, require only variable name- dynamically typed by internally.
10. **Both ->Procedural and Object Oriented language.**
11. Extensible->already existing program -> native language support.
12. Embedded-> we can use python program in any other language.
13. Python is not suitable for large scale enterprise application.

## **Where we can use python:**

1. Desktop application.
2. Networking application.
3. Games
4. Data Analysis
5. Data Science with python
6. Machine Learning
7. Artificial Intelligence (AI) application
8. IOT application

## **Flavors for python(Standard):**

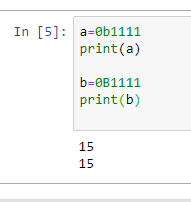
1. Cpython
2. Jython or JPython(work with Java language)
3. IronPython(work with C# language)
4. Pypy -> JIT(just in time)-> performance improvement.
5. Ruby python->(for ruby language)
6. Anaconda Python-> Large values of data processing.
7. Stackless - > (Python for Concurrency-> multithreading)

## **Identifiers (variable, class, method names):**

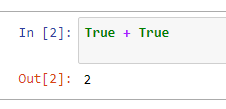
1. Rules to defined identifiers in python-> alphabet sysmbos(support Uper case and Lower case)
2. Digits( from 0 to 9), Underscore( \_ ) , special char not support($@# etc).
3. ( \_ ) single underscore 🡺 private (start with single underscore) ex. \_a, \_b
4. (\_\_) double underscore 🡺 strongly private (start with single underscore).
5. ( \_\_main\_\_ ) start and end with double underscore 🡺 language specified identifier defined by python.
6. TOTAL and total both are different identifier.
7. No length limit to defined identifiers ( aaaaaaaaaaaaaaaaaaa…..n)
8. 33 reserved words.(raise,assert if ,for. Etc)
9. **Long type is not there in python-3 until python-2 was there.**
10. **In python everything is Object. Like – init,float,bool,str,list,set etc**.
11. Any number which is start with – 0b or 0B ( zero and small or capital B) are called Binary format .

Example: a=0b1111 🡺values is 15

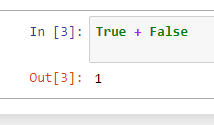
a=0B1111🡺15



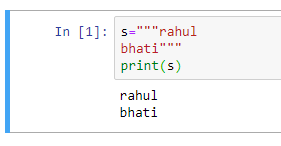
1. True + True = 2



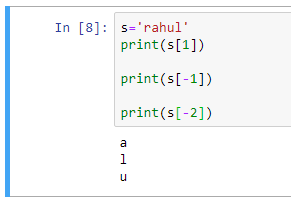
1. True + False = 1 (note🡺 true=1 and false= 0)



1. Second line 3 single or double quotes- Multiple line string literals



1. Positive and Negative index is possible in python. Means positive index is for Forward and negative index is for Backward, indexing moving from both direction.



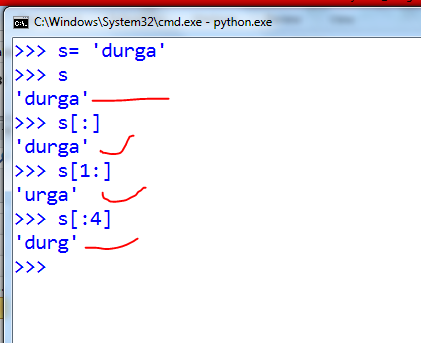
## **Slice Operator:**

Return substring from string or list – by Begin index and End index

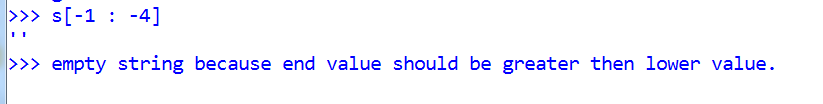
>>>s= “durga”

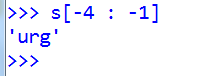
**Syntax -** varName[Begin : End]

1. s[ : ] 🡺 Full String
2. s[1:] 🡺 End is optional(default is length of the string)
3. s[:4] 🡺 Begin is optional(default is 0 index of the string)

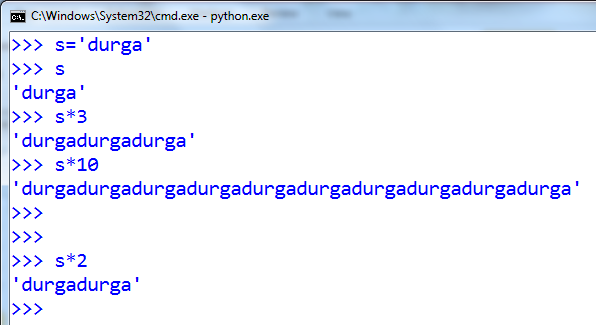


Note:

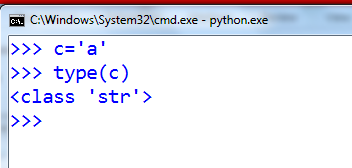




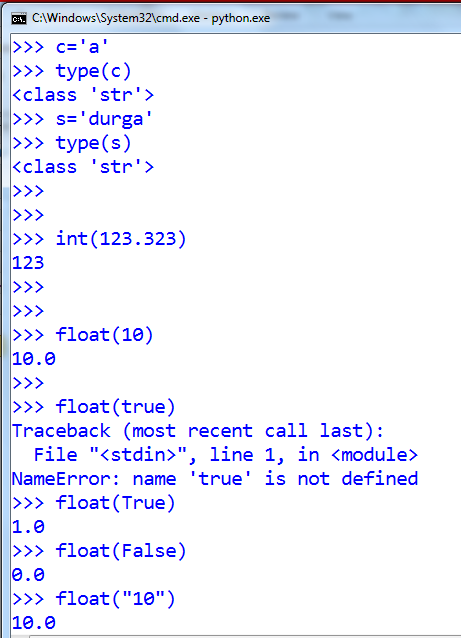
## **String Multiplication (\*):**

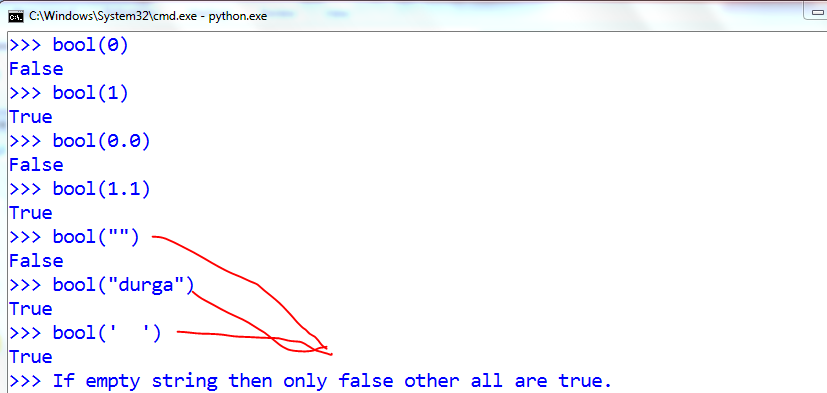


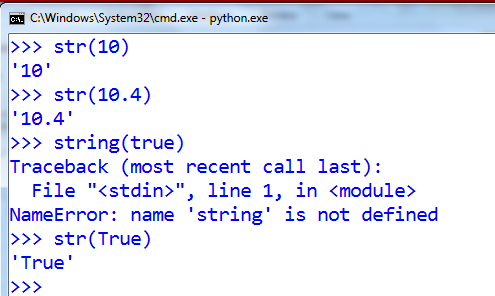
Note: char type is not there in the python.



## **Type checking (Variable type) and Type Casting:**







## **Data Types:**

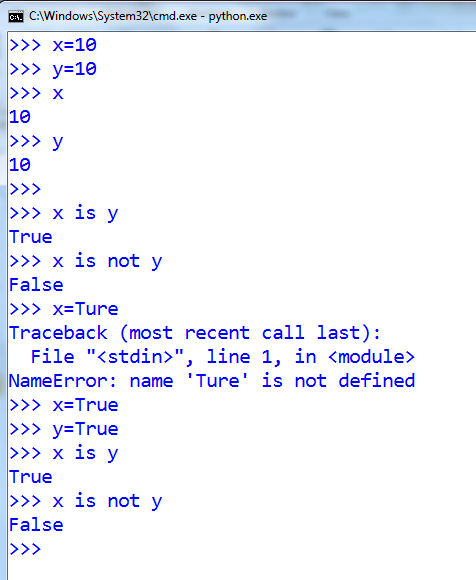
1. Inbuilt data types are:
2. Int
3. float
4. complex – (a+bj) where a=real part , b=imaginary part
5. bool
6. str
7. Sequence or collection related data types are:
8. bytes
9. bytearray
10. range (function and variable also – both)
11. list
12. tuple
13. set
14. frozenset
15. dict
16. none

## **Immutable & Fundamental Data Types: All objects in python are immutable.**

>>>> X=10

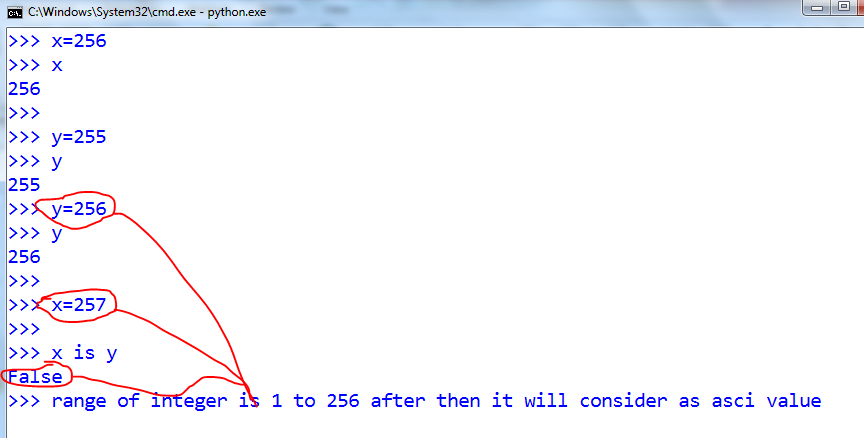
>>>> Y=20

Only one object will be create because both pointing to same **content.**



At the time of python interpreter starts:

From 0 to 256 integer object will be created at the beginning.



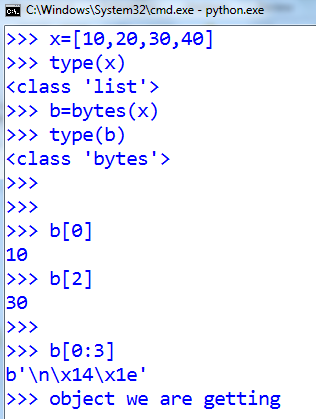
Only in the following range:

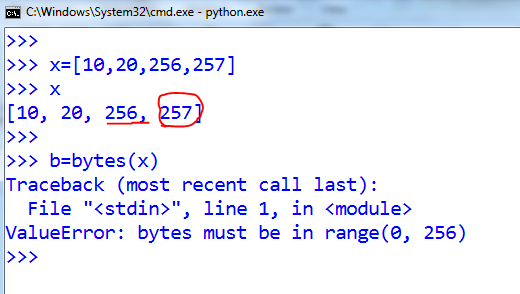
Int =🡺 0 t0 256

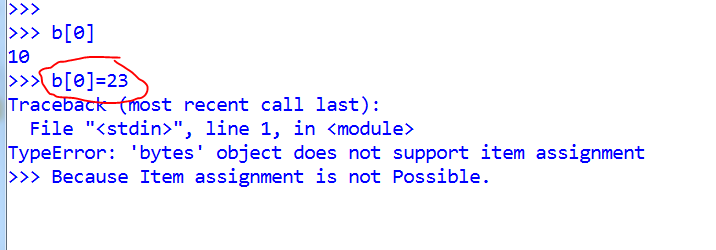
Bool 🡺 always

Str=🡺always

**Byte and ByteArray range also form 0 to 256**

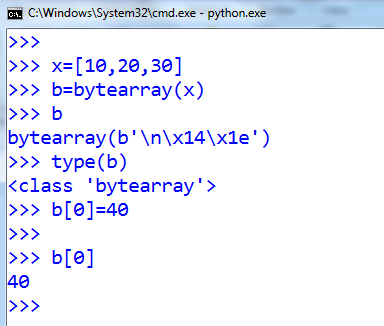


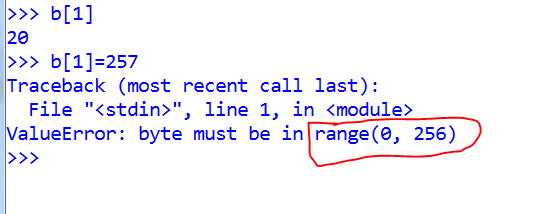




Because it’s an Immutable.

Byte and Bytearray both are same only difference is byte is immutable and bytearray is mutable





### **List**: duplicate are allowed no range limit any data type we can store.

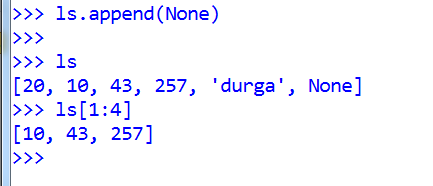
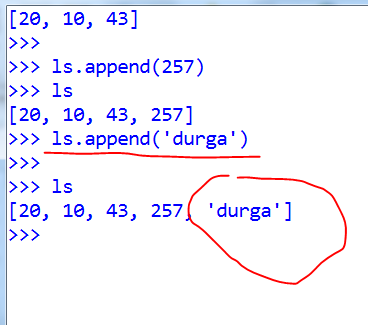
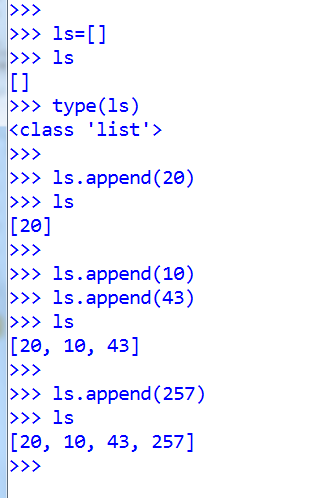
**Order preserved.**

**Duplicate allowed**

**Heterogeneous allowed**

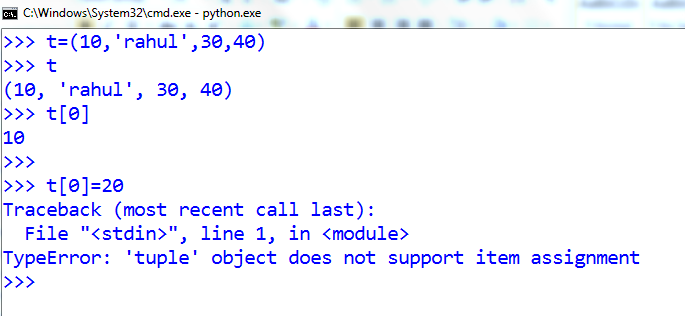
**Growable**

**Values should be enclosed with [].**



### **Tuple**: Immutable we cannot change their value once it declare.

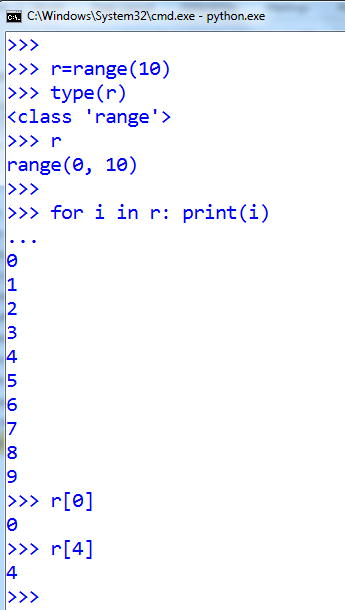
**List is mutable even Tuple is immutable, other than all same.**

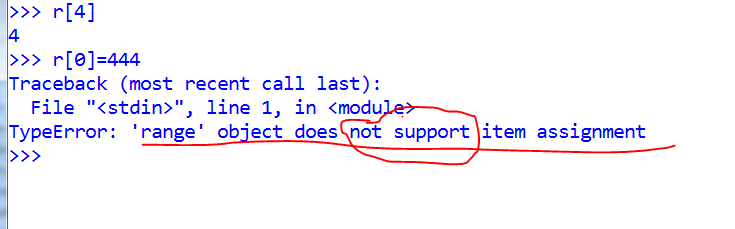
**Start and close with “( )”**

### Range (): range data type represents a sequence of values immutable.

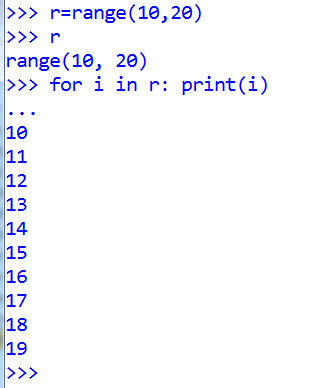
1. Form-1: range(end)

It represents values from 0 to 9.

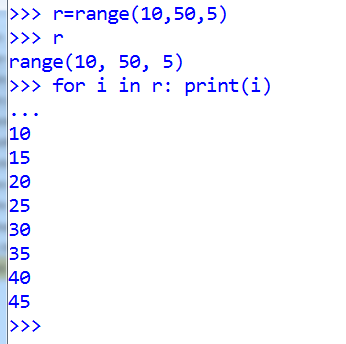


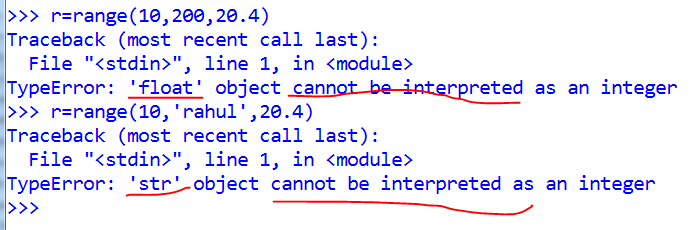


1. Form-2: start and end range (end will be number -1), means range(10,20).



1. Form-3: range(10,50,5) -> step 🡺 increment with 5.

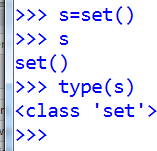


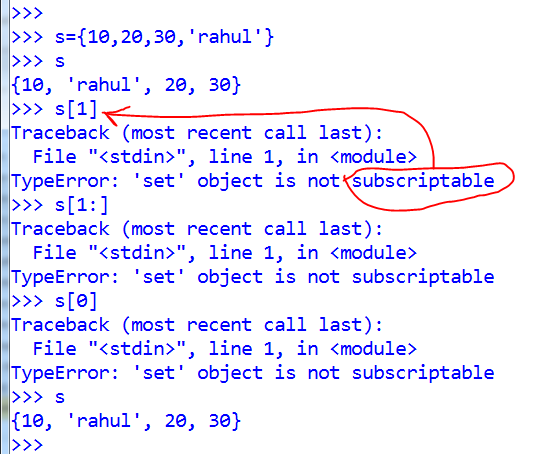
Range work with integer only Integer:

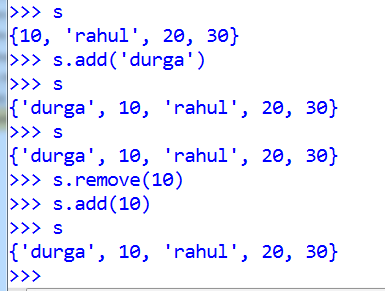
### Set: unlikeness only : does not support indexing, use { } .

It is mutable, slicing is not there.

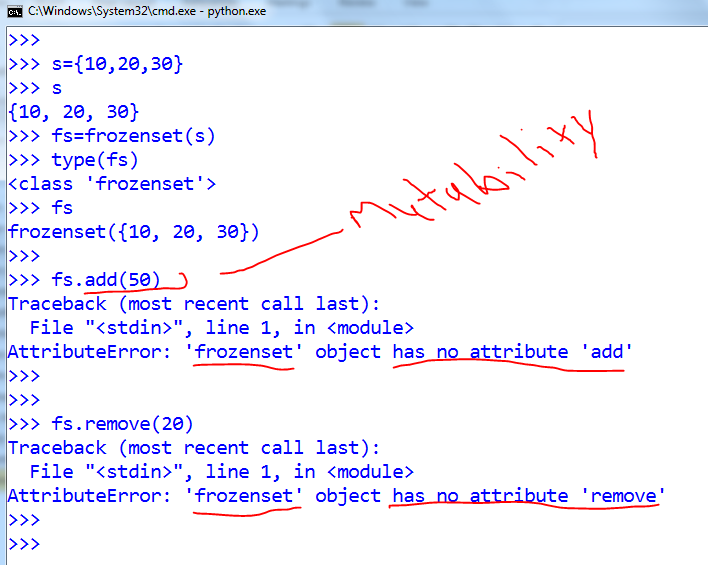
S=set()



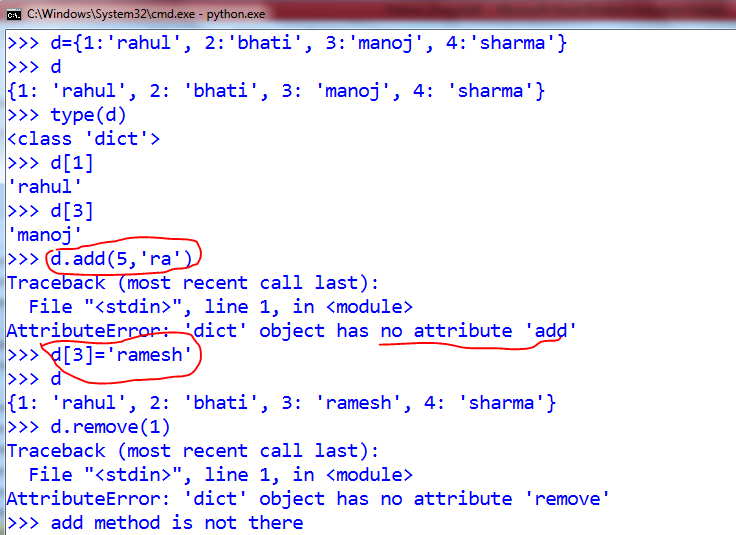




### Frozenset data type: same as set but only different is immutable.



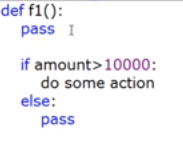
### Dirt🡺 ( directory ) = key and value pair. Group of individual object.



### No data type🡺

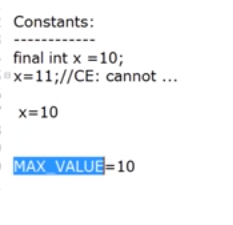
If you don’t want to do anything inside the block –

Just pass my code don’t do anything .

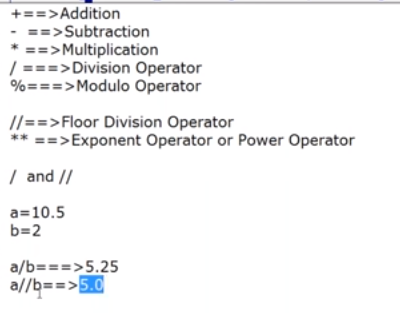


Final keyword is not there in python you cannot set final value.

But still you can declare as the MAX\_VALUE=10

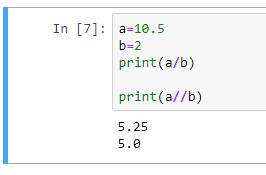


## Operators:



/ (division operator) 🡺 result always float type.

// (floor division) 🡺 If both argument are float type then result will be float type. Same as for int.



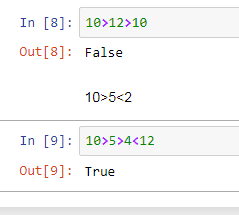
## **Relational Operator:**

Multiple comparison operator(chaining of relational operator)🡺

Like -🡪 10>20>30

If all conditions are true then only condition is true .

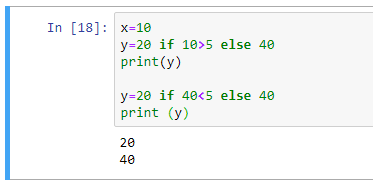
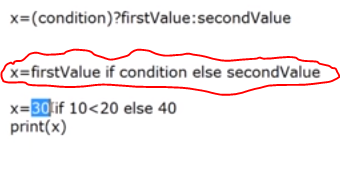
If at least one condition is false then condition is false.



Note: Increment(++x or x++ ) and decrement (-- x or x-- )operator is not there in python.



## **Ternary Operator:**



## Special Operator:

1. Identity Operator
2. Membership Operator

### Identity Operator🡺 “ is and is not“ we use.

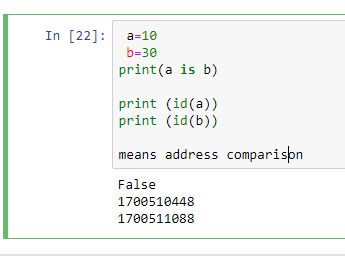
Is

Is not

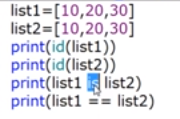
a = 10

b=30

if a is b 🡺 means address comparison

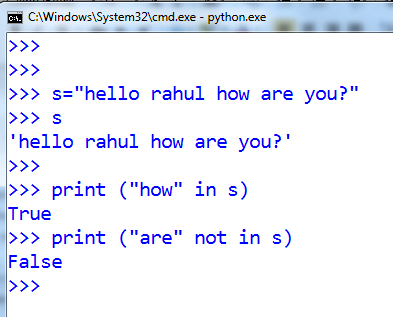


**Note “ == “ double equals is for content comparison**



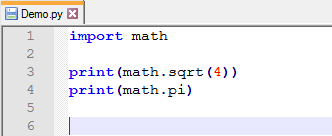
### Membership Operator: To check whether given item is present in the list or not.

There are two type: “ In “ and “ not in”

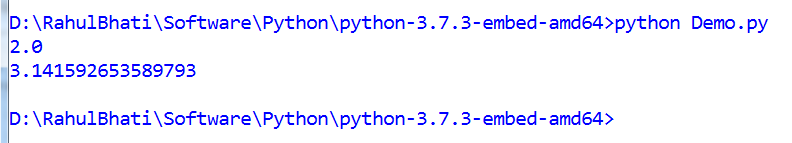


# **Module: module are collection of Variables, class and functions. Like libraries and API in java.**

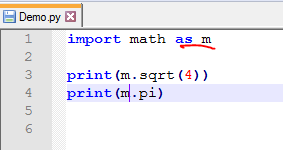
We need to import modules to use it.



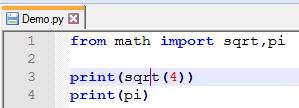
Output:



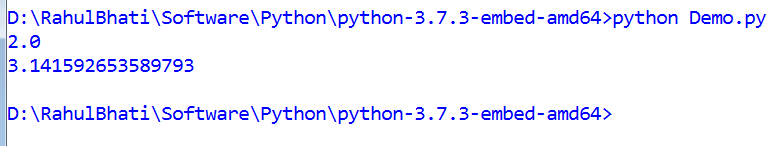
## **Aliasing:**



To import particular class, method, variables:



Output:



# **Input and Output Statement:**

## **Input Statement:**

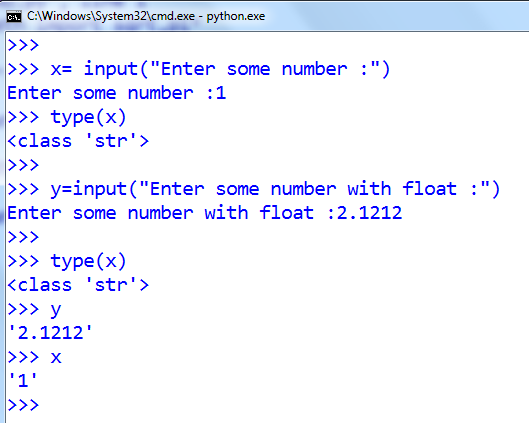
Input: read dynamic data from keyboard

There are 2 types functions are available in python.

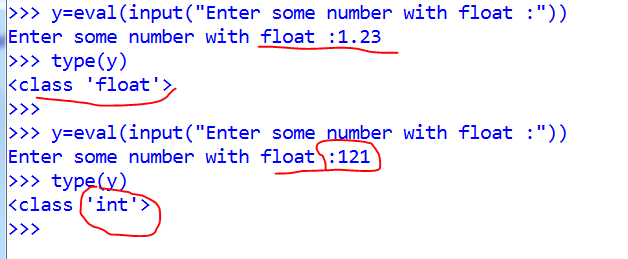
1. In python-2 –
   1. row\_input(“We require to use typecasting”)
   2. input(“Integer type only”)

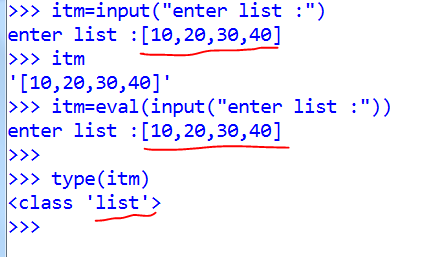
**But in python -3 : only once function is there:**

Input(“any data type but we need typecasting”)



### Eval function: (eval()) : Internally perform type casting.

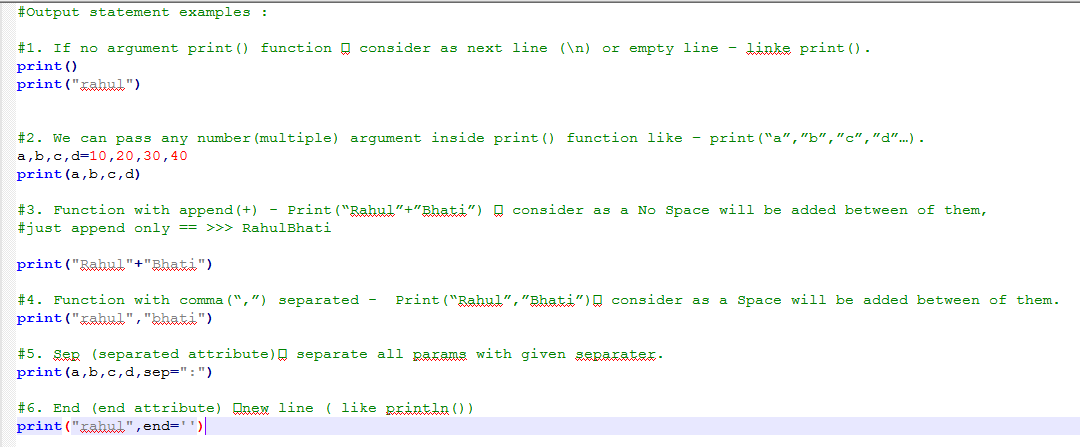




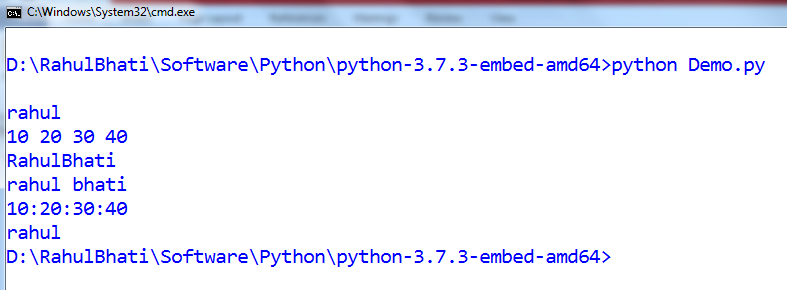
## **Output Statement (print()):**

Python provide a function for output statement called 🡺 print()

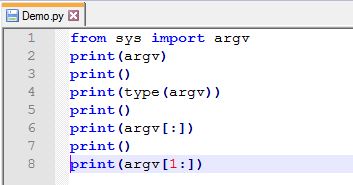
1. If no argument print() function 🡺 consider as next line (\n) or empty line – like print().
2. We can pass any number(multiple) argument inside print() function like – print(“a”,”b”,”c”,”d”…).
3. Function with append(+) - Print(“Rahul”+”Bhati”) 🡺 consider as a **No Space** will be added between of them, just append only == >>> RahulBhati
4. Function with comma(“,”) separated - Print(“Rahul”,”Bhati”)🡺 consider **as a Space** will be added between of them.
5. Sep (separated attribute)🡺 separate all params with given separate.
6. End (end attribute) 🡺new line ( like println())



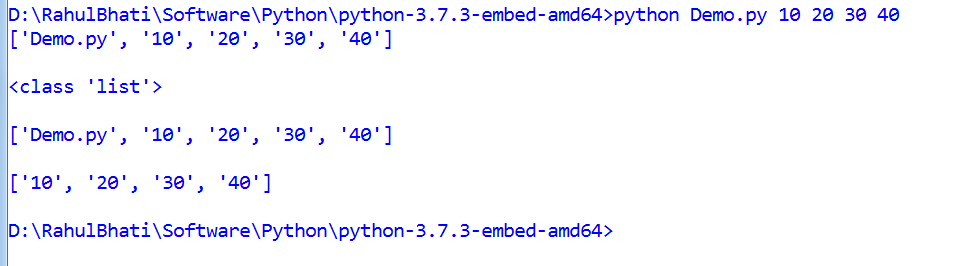
Output:



# **Command Line Arguments: (need to import sys) 🡺 argv**



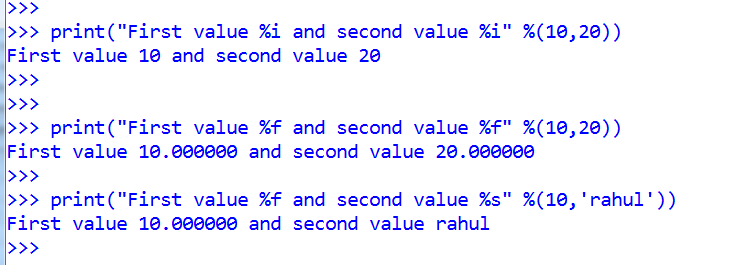
Output: **with passing arguments:**



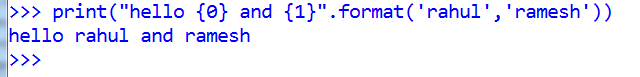
## **Type Characters:**

1. %i 🡺 int type
2. %d 🡺 int type
3. %f 🡺 float type
4. %s 🡺 str type

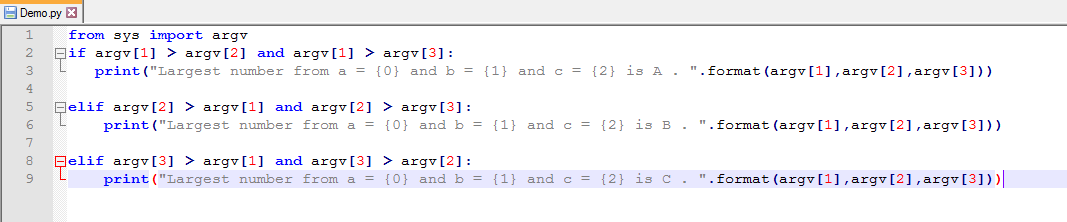
Syntax 🡺 print(“formatted string” %i(variable list))



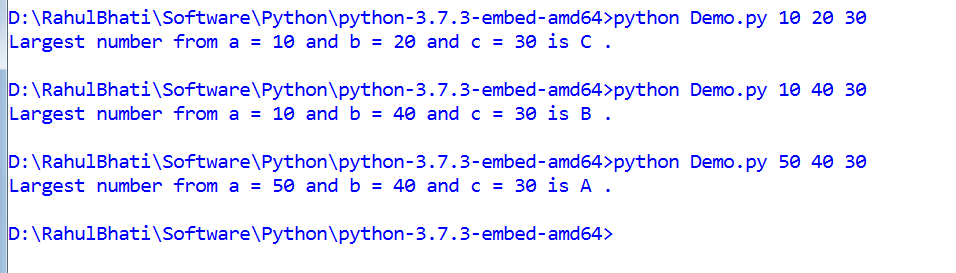
## **Replacement Operator ( {} ):**



# **Conditional / Selection Statement:**



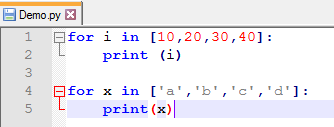
Output:



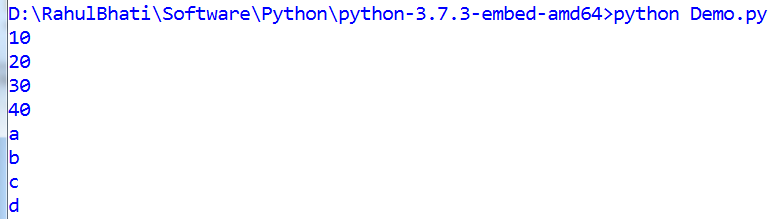
**Note: Switch … Case statement not supported by Python.**

# **Loop Statement: python support two types loop.**

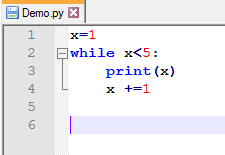
1. For loop :



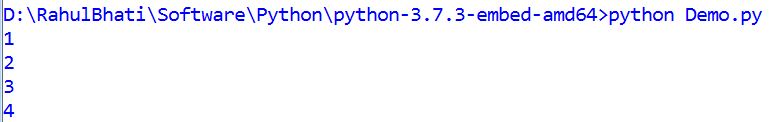
Output:



1. While loop:



Output:



# **String:**

🡺We can access characters from string :

1. By using Index.
2. By using Slice operator.

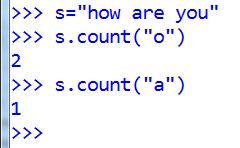
Both positive and negative index we can use in python(forward and backword)

🡺Istrip() 🡪 remove start space from the string

🡺rstrip() 🡪 remove end space from the string.

🡺trip() 🡪 remove space from both side.

🡺count() 🡪 count substring from the give string.



🡺find(substring) 🡪 find substring from give string.

🡺index()

🡺rfind()

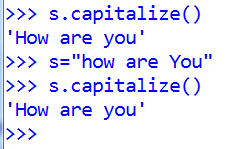
🡺replace()

🡺split()

🡺join()

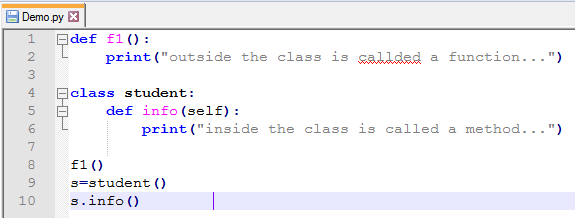
🡺upper(),lower()

🡺capitalize() 🡪 Only first start capital remaining same.

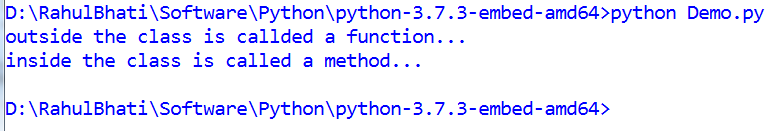


# **Method and Functions:**

If we declare methods outside of the class is called function and inside a class is called method.



Output:



# Pip install :



