# Python Language

Father of python ***'GUIDO VAN ROSSUM'*** in 1980.It was originally released in ***1991 version-0.9.0***,then version-***2.0 in 2000***,then version-***3.0 in 2008***.

- Python is high level genral purpose programing language(GPL)
- OPPOSITE OF GPL- Domain sepefic purpose programing language(DSL).
- Python is casesensetive.
- Dynamically typed. (we need to tell python data type).

## Comments

- Comments in python are identified with a hash symbol(#)and extend to the end of the line.

## Docstring

- Python docstring is a string ['''] used to document a python module,class, function ,so programmers can understand what it does without having to read details of the implementation.

## Print function (giving output)

- The Print()function prints the specified message to the screen or other standaed output device. syantax - print()

```
print('name' ) # print funtion
```

## Data type

- There are two types of data

- ## Primitive
- Number - int,fload,comlex,boolean(bool)
- String
- Non

- ## Non-primitive

```
print(22) # integer
print(2.02) # fload

print(34j) # complex

print(True) # bool

print('rohit') # string
```

## Non-primitive

```
  - tuples
  - sets
  - dictionaries
  - lists
```

- non-primitive data types are also called Data Structures.
- In Data Structures primitive data types can be stored in a sequence or structure.

```
# list                                    # [indexing]
x = ["str",9.9,[2,3,4],"rohit"]
x[3]

X = ["ROHIT",["mohit"]]
X[-1]

# list indexing


x = ['str', 'hello',True ,[1,2,[1,2,3,4,5,6,7,[2,3,4,5,6,7]]]]
x[3][2][7][2]
```

## len.

```
x = ['str', 'hello',True ,[1,2,[1,2,3,4,5,6,7,[1,2,3,4,5,7]]]]

print(len(x))

T = [1,2,3,4,5,6,7,8,9,['rishi',1,2,3,['rishi',3,4,
['hello','world']]]]
print(len(T))

x[3]

x[2]

# List.append(item)  - add a single item at end of the list

x = [1,2,3,4,5,6,7,8]
```

```
x.append(123)
x.insert(3,45)      # insert takes 2 agruement first index num second
u and to insert
print(x)

# .extend(iterable)
# iterable - string , list, tuples , set , dictionary


lst = [1,2,3]
lst.extend(["hello"])
lst


x = [1,1,2,2,3,3,4,4]
x.extend(["555555",["rohit"]])
x

# remove()
lst = [1,4,2,7,5,6,7]
lst.remove(7)
lst.remove(7)
lst
```

```
# .count
x = [1,2,3,4,4,4]
x.count(4)
```

sort()

lst.sort() lst

```
x = [2,5,4,6,8,7,5,4,3]
x.sort()
print(x)

 #.reverse()
x.sort(reverse=True)
print(x)

r = x.sort(reverse=False)
print(x)

x = ["rohit","mohit", 12345 , 6586867]

x.reverse()
```

```
print(id(x))
print(id(x.reverse))

x = [2,3,4,5,7.3,6,5,4,4,3,5,6,7,8]
y = x.count(4)
x.reverse()
print(x)
print(y)


# .split                 # will give us in output list

# Tuple


x = ("str",)
type(x)

y = 1,1                   # tuple
y

# .count( )
tup_1 = (1,2,5,3,2)
tup_1.count(2)

# .index()
tup_1.index(5)
```

## Data types

- mutable
- immuatable # Object whose value is unchangeable once they are created are called immutable.

## Mutable

## An object that allows you to change its values without changing its identity is a mutable object.

## Every object has an identity (Address,memory,location),a type and a value.

- All data in a Python program is represent=ented by objects.
- Every object has an identity ,a type and a value.
- An object type determines the operations that the objects supports and also defines the possible value for objects of that type.
- The value of some objects can be changed .objects whose value can change without changing its identity are said to be mutable.
- Object whose value is unchangeable once they are created are called immutable.

```
x = 12          # x variable is not a contaner its just assining or pointing to object,12 the value thats why its keep changing
print(id(x))
x = 13
print(id(x))

x = [13]
y = [14]
x =[12]
print(id(x))
print(id(y))

x

x = 'data'
print(id(x))
x = "science"
print(id(x))

print(x)

y =[23,43]
print(id(y))
y = [67,78]
print(id(y))
```

```
print (y)

p = "My self data sientist"      # immutable
print(id(p))
p.replace(" ",":")
print(id(p.replace))

z = [1,2,3,4,5]             # mutable  -- THE ID OR LOCATION OR ADDRESS
OF A LIST NEVER cHANGES EVEN IF THE SAID LIST IS ELEMENT EDITED
print(id(z))
print(z.reverse())
print(id(z.reverse))
z[4]=55
id(z.reverse)

 # mutable # MUTABLE-- THE ID OR LOCATION OR ADDRESS OF A LIST
# NEVER CHANGES EVEN IF THE SAID LIST IS ELEMENT EDITED.


list = [12,13,14,66,"data"]
print(id(list))
z = list.append("scientist")
print(list)
print(id(list))
print(id (z))
print(z)


p = "rohit mohit"
print(id(p))
p.replace(" ","5")
print(p)
print(id(p.replace))

p

lo = 'rishi'
print(id(lo))

print(id(z))
```

## Type function

- Type function is to know the deta types.
- syantax - type().

```
type(45)

type('hello')
```

## Type casting

- type casting is to convert one data type to other data type.

```
print(int(56.09)) # float to int

print(int('56')) # str to int
```

## Variables

- Variable is a name where we stores a value or data.

## Rules of variable

- A variable name must start with an alphabet or an underscore.
- A variable name cannot start with a number or any special character (@,#,%,&)etc.
- A variable name can only contain alphabets number and underscores.
- The use of special charactors is not allowed.
- Variable name are casesensitive in python.
- Python keywords cannot be used as variable names.

```
num = 23
print(num)
x = 'rishi'
print(x)
_class = 44
print(_class)

_3 = "rohit"
print(_3)
```

## Espace sequence

- ### What is espace sequence
- Character combinations consisting of a backslash()following by a letter or a combination of digits are called escape sequences.

- 1- (\') -Single code.

- 2- (\b) -Backspace.

- 3- (\) -Backslash.

- 4- (\n) -New line correctotr.
- 5- (\t) -Tab.
- 6- (\r) -carreiage return.