In [1]:
```
print("Hello World!")
print("Welcome in Python Programmming!!")
```

```
Hello World!
Welcome in Python Programmming!!
```

In [1]:
```
#veriable take space temporary in memory
student="Raj"
```

In [2]:
```
student
```

Out[2]:
```
'Raj'
```

In [3]:
```
student=121
```

In [4]:
```
student
```

Out[4]:
```
121
```

In [5]:
```
student="Anuradaha Kumari"
```

In [6]:
```
student
```

Out[6]:
```
'Anuradaha Kumari'
```

In [1]:
```
#Veriable Uses in program

a1=10
a1
```

Out[1]:
```
10
```

In [2]:
```
a1=3.14
a1
```

Out[2]:
```
3.14
```

In [3]:
```
type(a1)
```

Out[3]:
```
float
```

In [5]:
```
name="Rindu"
name
```

Out[5]:
```
'Rindu'
```

In [6]:
```
type(name)
```

Out[6]:
```
str
```

In [10]:
```
name="sima"
name
```

Out[10]: 'sima'

In [11]:
```
roll=121
roll
```

Out[11]: 121

In [12]:
```
type(roll)
```

Out[12]: int

roll=true roll

In [13]:
```
type(roll)
```

Out[13]: int

In [14]:
```
roll=True
roll
```

Out[14]: True

In [15]:
```
Roll=False
roll
```

Out[15]: True

In [16]:
```
Roll
```

Out[16]: False

In [17]:
```
type(Roll)
```

Out[17]: bool

In [18]:
```
#Complex data type

comp=3+4j
comp
```

Out[18]: (3+4j)

In [19]:
```
type(comp)
```

Out[19]: complex

In [ ]:
```
#operater in python😎
#airthmetc operater
#logical operater
#bitwise operater
#conditional operater  //Turnary operater
#comperision operater
```

In [21]:
```
#.1 Airthmetic operater (+,-,*,/)

a=10
b=25
a,b
```

Out[21]: (10, 25)

In [22]:
```
sum=a+b
sum
```

Out[22]: 35

In [23]:
```
sum=b-a
sum
```

Out[23]: 15

In [24]:
```
sum=b/a
sum
```

Out[24]: 2.5

In [25]:
```
sum=a*b
sum
```

Out[25]: 250

In [26]:
```
sum=a-b;
sum
```

Out[26]: -15

In [ ]:
```
#relational operater (<,>,!=)
```

In [27]:
```
a>b
```

Out[27]: False

In [28]:
```
b<a
```

Out[28]: False

In [29]:
```
b>a
```

Out[29]: True

In [30]:
```
a!=b
```

Out[30]: True

In [31]:
```
b<=a
```

Out[31]: False

In [32]:
```
a==b
```

Out[32]: False

In [ ]:
```
#Logical operater (&,||,)
```

In [33]:
```
a&b
```

Out[33]: 8

In [35]:
```
b|a
```

Out[35]: 27

In [36]:
```
b&b
```

Out[36]: 25

In [37]:
```
a|a
```

Out[37]: 10

In [38]:
```
a=True
b=False
```

In [39]:
```
a|a
```

Out[39]: True

In [40]:
```
b&a
```

Out[40]: False

In [41]:
```
b==b
```

Out[41]: True

In [42]:
```
a|b
```

Out[42]: True

In [43]:
```
# Python String 😀
```

In [45]:
```
str1='This is my first string'
str1
```

Out[45]:
```
'This is my first string'
```

In [46]:
```
str2="This is my second string!"
str2
```

Out[46]:
```
'This is my second string!'
```

In [49]:
```
str3='''
This is string
lots of line
three quatation
yes
'''
```

In [50]:
```
str3
```

Out[50]:
```
'\nThis is string\nlots of line\nthree quatation\nyes\n'
```

In [51]:
```
my_string="This is string"
```

In [52]:
```
my_string
```

Out[52]:
```
'This is string'
```

In [53]:
```
my_string[0]
```

Out[53]:
```
'T'
```

In [54]:
```
my_string[-1]
```

Out[54]:
```
'g'
```

In [57]:
```
my_string[5:10]
```

Out[57]:
```
'is st'
```

In [58]:
```
#string length find strlen()
my_string
```

Out[58]:
```
'This is string'
```

In [59]:
```
len(my_string)
```

Out[59]:
```
14
```

In [60]:
```
my_string.lower()
```

Out[60]: 'this is string'

In [61]:
```
my_string.upper()
```

Out[61]: 'THIS IS STRING'

In [63]:
```
my_string.replace('is','are')
```

Out[63]: 'Thare are string'

In [64]:
```
# Repitation of the number are printing for
str_new="welcome in lovely professional university 100 100 100"
```

In [65]:
```
str_new
```

Out[65]: 'welcome in lovely professional university 100 100 100'

In [66]:
```
str_new.count('100')
```

Out[66]: 3

In [67]:
```
# for finding location of the particular character

my_string="Hello world is awsome"
```

In [69]:
```
my_string.find('i')
```

Out[69]: 12

In [72]:
```
str_final="prsident_kovind is best president of india!!"
```

In [74]:
```
# for spliting the character in the string
str_final.split('e')
```

Out[74]: ['prsid', 'nt_kovind is b', 'st pr', 'sid', 'nt of india!!']

In [75]:
```
# Besic data structure in python: -(Tuple,List,Dictionary,set) 4 Types  //
```

In [78]:
```
# 1.tuple
tup1=(1,True,3.14,5-2j)
tup1
```

Out[78]: (1, True, 3.14, (5-2j))

In [79]:
```
type(tup1)
```

Out[79]: tuple

In [97]:
```
#for invidual element extracting  //
tup1[3]
```

Out[97]: 4

In [87]:
```
# can't change the value of touple value becouse it is immutable
# tup1[2]="this is car"
```

In [88]:
```
len(tup1)
```

Out[88]: 4

In [92]:
```
tup1=(1,2,3,4,5)
tup2=(6,7,8,9,10)
tup1,tup2
```

Out[92]: ((1, 2, 3, 4, 5), (6, 7, 8, 9, 10))

In [93]:
```
tup1 + tup2  # for concatinating the two touple
```

Out[93]: (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

In [94]:
```
tup1*5    # for repating same data
```

Out[94]: (1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5)

In [95]:
```
tup2*4
```

Out[95]: (6, 7, 8, 9, 10, 6, 7, 8, 9, 10, 6, 7, 8, 9, 10, 6, 7, 8, 9, 10)

In [99]:
```
tup1*4+ tup2
```

Out[99]: (1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

In [100]:
```
# for finding minimum values or max using min/max function

min(tup1)
```

Out[100]: 1

In [101]:
```
max(tup2)
```

Out[101]: 10

In [4]:
```
# List data structure : - its working on []  square brakets and it is a mutabl
e #

list_first=[1,'Raj',3.14,True]

list_first
```

Out[4]: [1, 'Raj', 3.14, True]

In [5]:
```
type(list)
```

Out[5]:
```
list
```

In [6]:
```
list_first[0]
```

Out[6]:
```
1
```

In [7]:
```
list_first[1:3]
```

Out[7]:
```
['Raj', 3.14]
```

In [8]:
```
# change value
list_first[0]=100
list_first
```

Out[8]:
```
[100, 'Raj', 3.14, True]
```

In [9]:
```
# for append value

list_first.append(-87.99)
list_first
```

Out[9]:
```
[100, 'Raj', 3.14, True, -87.99]
```

In [11]:
```
# for deleting the element
list_first.pop()
```

Out[11]:
```
-87.99
```

In [12]:
```
list_first
```

Out[12]:
```
[100, 'Raj', 3.14, True]
```

In [13]:
```
l1=[121,'12',90,True,False,500]
l1
```

Out[13]:
```
[121, '12', 90, True, False, 500]
```

In [14]:
```
l1.reverse()   #for reverse the element
```

In [15]:
```
l1
```

Out[15]:
```
[500, False, True, 90, '12', 121]
```

In [16]:
```
# for inserting the element in the list
l1.insert(1,'Hello')
```

In [19]:
```
l1
```

Out[19]:
```
[500, 'Hello', False, True, 90, '12', 121]
```

In [23]:
```
l2=['a','f','d','x','z']
l2
```

Out[23]:
```
['a', 'f', 'd', 'x', 'z']
```

In [24]:
```
# for soting the element
l2.sort()
```

In [25]:
```
l2
```

Out[25]:
```
['a', 'd', 'f', 'x', 'z']
```

In [26]:
```
l1+l2  # conacatinate the two list
```

Out[26]:
```
[500, 'Hello', False, True, 90, '12', 121, 'a', 'd', 'f', 'x', 'z']
```

In [31]:
```
l2*2
```

Out[31]:
```
['a', 'd', 'f', 'x', 'z', 'a', 'd', 'f', 'x', 'z']
```

In [32]:
```
l1*2+l2
```

Out[32]:
```
[500,
 'Hello',
 False,
 True,
 90,
 '12',
 121,
 500,
 'Hello',
 False,
 True,
 90,
 '12',
 121,
 'a',
 'd',
 'f',
 'x',
 'z']
```

In [33]:
```
# Dictionary data structure  unordered collection of key value
# it is a key value and pair and its define by {} ,
d1={'Apple':110,'Mango':90,'Banana':50,'Lichi':90}
```

In [34]:
```
d1
```

Out[34]:
```
{'Apple': 110, 'Mango': 90, 'Banana': 50, 'Lichi': 90}
```

In [35]:
```
type(d1)
```

Out[35]:
```
dict
```

In [36]:
```
d1.keys()
```

Out[36]:
```
dict_keys(['Apple', 'Mango', 'Banana', 'Lichi'])
```

In [37]:
```
d1.values()
```

Out[37]:
```
dict_values([110, 90, 50, 90])
```

In [38]:
```
# for modification in dictionary

d1
```

Out[38]:
```
{'Apple': 110, 'Mango': 90, 'Banana': 50, 'Lichi': 90}
```

In [40]:
```
d1['Orange']=70
d1
```

Out[40]:
```
{'Apple': 110, 'Mango': 90, 'Banana': 50, 'Lichi': 90, 'Orange': 70}
```

In [43]:
```
d1['mango']=80
d1
```

Out[43]:
```
{'Apple': 110,
 'Mango': 90,
 'Banana': 50,
 'Lichi': 90,
 'Orange': 70,
 'mango': 80}
```

In [44]:
```
d1
```

Out[44]:
```
{'Apple': 110,
 'Mango': 90,
 'Banana': 50,
 'Lichi': 90,
 'Orange': 70,
 'mango': 80}
```

In [46]:
```
# dictionary with function

d2={'watermelan':100,'grapes':90,'cocounut':40}
d2
```

Out[46]:
```
{'watermelan': 100, 'grapes': 90, 'cocounut': 40}
```

In [47]:
```
d1.update(d2)  #for updating dictionary
```

In [48]:
```
d1
```

Out[48]:
```
{'Apple': 110,
 'Mango': 90,
 'Banana': 50,
 'Lichi': 90,
 'Orange': 70,
 'mango': 80,
 'watermelan': 100,
 'grapes': 90,
 'cocounut': 40}
```

In [49]:
```
d1.pop('Banana')  #for deleting elemnet from dictionary
```

Out[49]:
```
50
```

In [50]:
```
d1
```

Out[50]:
```
{'Apple': 110,
 'Mango': 90,
 'Lichi': 90,
 'Orange': 70,
 'mango': 80,
 'watermelan': 100,
 'grapes': 90,
 'cocounut': 40}
```

In [51]:
```
# set 😎😃 -> is unordered and unindexed collection of element.// enclosed wi
th {}
#it doesnot contain duplicate element in a set //
```

In [52]:
```
s1={1,3.14,'Raj'}
s1
```

Out[52]:
```
{1, 3.14, 'Raj'}
```

In [53]:
```
s1={1,'Kumar','Raj'}
s1
```

Out[53]:
```
{1, 'Kumar', 'Raj'}
```

In [54]:
```
s1
```

Out[54]:
```
{1, 'Kumar', 'Raj'}
```

In [55]:
```
s1.add(False)
s1
```

Out[55]:
```
{1, False, 'Kumar', 'Raj'}
```

In [58]:
```
# Update element in set
s1.update(['Good',7856,5+3j])
s1
```

Out[58]: {(5+3j), 1, 7856, False, 'Good', 'Kumar', 'Raj'}

In [59]:
```
# for removing the element from the list
s1.remove('Good')
s1
```

Out[59]: {(5+3j), 1, 7856, False, 'Kumar', 'Raj'}

In [60]:
```
# SET function

s1={1,2,3,4,5}
s2={5,6,7,8,9}

s1.union(s2)   # for merginging two set in one set
```

Out[60]: {1, 2, 3, 4, 5, 6, 7, 8, 9}

In [65]:
```
s3={8,9,10,}

s1.intersection(s3)
```

Out[65]: set()

In [ ]:
```
# if condition in python
```

In [68]:
```
a=10
b=25
if b>a:
    print("B is greater then a")
```

```
B is greater then a
```

In [72]:
```
a=100
b=25
if b>a:
    print("B is greater then a")
else:
     print("a is greater then b")
```

```
a is greater then b
```

In [77]:
```
a=190
b=500
c=300

if(a>b) & (a>c):
    print("A is the greatest")
elif(b>a) & (b>c):
    print("B is the greatest")
else:
    print("C is the greatest")
```

```
B is the greatest
```

In [80]:
```
# if with tuple

tup1=('a','b','c')
if 'z' in tup1:
    print("Value is present in tup1")
else:
    print("Value is not present in tup1")
```

```
Value is not present in tup1
```

In [3]:
```
# take two number from the user and add this  using function
def sum():

    x=int(input("Enter first number :"))
    y=int(input("Enter second number :"))

    z=x+y
    print("Sum of two number is :",z)

    sum()
    sum()
```

In [ ]:
```

```