**Length of Tuple inside List**

list2 = [(1,2,3),('x','Y','Z','A')]

print (list2)

print (len(list2))

print (len(list2[1]))

**Length ,Max, Min of the Tuple (based on Ascii value)**

tup1 =(1,2,3,'x','y','m')

tup2 =('x','y','m','N')

print (len(tup1))

print (max(tup2))

print (min(tup2))

**Multiplication**

tup1 =(1,2,3,'x','y','m')

print (tup1 \*2)

print ('x' in tup1)

**Example 2 (Multiply one element)**

tup1=(10,20,30)

tup2=tup1 [:2]

tup3=tup1[2:]\*3

tup1 = tup2+tup3

print (tup1)

**Del Tuple**

tup1 =(1,2,3)

tup2 =(4,5,6)

tup3 = tup1+tup2

print (tup3)

del (tup2)

print (tup2)

**Update List inside Tuple**

tup2 = ([5,6,7],[10,11,12],13,14)

print (tup2)

print (type (tup2))

print (type (tup2[0]))

tup2 [1] = [10,15,14] – Error

tup2[1][1] = [10,15,14]

print (tup2)

**Convert List to Tuple**

Tup2 = (10,11,12)

List3 = list (Tup2) – Error

Tup5 = tuple (List3)

print (Tup2)

print (List3)

print (Tup5)

**String Operations**

x = 'Good Day'

print (len (x))

print (x [2])

print (x [2:8])

print ('o' in x)

x = ('G','o','o','d','D','a','y')

print (len (x))

print (x [2])

print (x [2:8])

print ('o' in x)

**% s**

a = 'Python'

b = 'Training'

print ('Welcome to %s %s' % (a,b))

a = 'Python '

print (a \* 2)

aa = a [0:10] + 'Training'

print (aa)

**Find Position D**

str1 = 'Good Day'

print (str1.find('D'))

**Example Set**

x = set ('Good Morning')

print (x)

**Union in Set**

a = {1,2,3,4}

b = {3,4,5,6,8}

c = a|b

print (c)

print (a&b)

print (a-b)

**Dictionary**

person = {'First\_Name': 'John', 'Last\_Name': 'Sam','Salary' : 5000}

print (person['First\_Name'])

person = {'first\_name':['John','Sam','Jim','Kim'],'Last\_name': 'Len'}

print (person['first\_name'])

print (person['first\_name'][2])

person = {'first\_name':['John','Sam','Jim','Kim'],'Last\_name': 'Len'}

print (person['first\_name'])

print (person['first\_name'][2])

person['first\_name'][2] ='XYZ'

print (person['first\_name'])

**Del from Dictionary**

person = {'first\_name':['John','Sam','Jim','Kim'],'Last\_name': 'Len'}

print (person['first\_name'])

del (person['first\_name'][2])

print (person['first\_name'])

**Clear Dictionary**

person.clear ()

**Dictionary Keys , Values**

person = {'first\_name':['John','Sam','Jim','Kim'],'Last\_name': 'Len'}

print (person.keys ())

print (person.values ())

print (person.items())

print (person.get ('first\_name'))

dict1 = {'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}

double\_dict1 = {k:v\*\*2 for (k,v) in dict1.items()}

print(double\_dict1)

**Addition New column**

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(thisdict)

thisdict["color"] = "red"

print(thisdict)

**Pop**

thisdict = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
thisdict.pop("model")  
print(thisdict)

**Copy Dictionary**

thisdict = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
mydict = thisdict.copy()  
print(mydict)

Nested Dictionary

myfamily = {

"child1" : {

"name" : "Emil",

"year" : 2004

},

"child2" : {

"name" : "Tobias",

"year" : 2007

},

"child3" : {

"name" : "Linus",

"year" : 2011

}

}

print (myfamily)

print ('\n Values \n')

print (myfamily.values())

print ('\n Items \n')

print (myfamily.items())

**Nested Dictionary Example 2**

myfamily = {'Father': {'Name':'Tom', 'Age':60},

'Mother':{'Name':'Ann', 'Age':55},

"Children": {

'Child1': {'Name':'Eric', 'Age':22},

'Child2': {'Name':'Jen', 'Age':18}

}

}

print (myfamily)

print(myfamily['Father'])

print (myfamily['Children'])

print (myfamily['Children']['Child1'])

**Loop Example**

customer\_name = ['John','Sam','Rick','Jim', 'Ken']

order\_id = ['AA123','AAA110','BBB111','BBB123','CCC100']

Hello John Your order id is AA123

Hello Sam Your order id is AAA110

Hello Rick Your order id is BBB111

Hello Jim Your order id is BBB123

Hello Ken Your order id is CCC100

**Ans:**

for (c,o) in zip (customer\_name,order\_id ):

print ('Hello',c,' Your order id is',o)

**Python zip() Function**

The zip() function returns a zip object, which is an iterator of tuples where the first item in each passed iterator is paired together, and then the second item in each passed iterator are paired together etc.

The zip() function **combines the contents of two or more iterables**. zip() returns a zip object. This is an iterator of tuples where all the values you have passed as arguments are stored as pairs. Python's zip() function takes an iterable—such as a list, tuple, set, or dictionary—as an argument.

**Example zip() Function**

city = ['Chennai', 'Mumbai', 'Delhi', 'Kolkata']

population = [5000, 8000, 10000, 7500]

new\_dict = {city: population for city,

population in zip(city, population)}

print(new\_dict)