Tuple manipulations

- Predefined class for tuple is <class 'tuple'>
- Heterogeneous Objects are allowed [dis-similar values]
- Insertion order is maintained
- tuple is Immutable
- Duplicate Objects are allowed
- None type Objects are allowed
- All Object in the tuple must be taken in between ( )
- Every Object must be separated by using ,

Creating tuple Collection:

```
#App-1 Creating an empty tuple
t1=()
print("Type is : ",type(t1))
print("Data is : ",t1)

#Creating tuple collection with Objects
print("-"*30)
t2=(10,"A",3.14,None,10)
print("Type is : ",type(t2))
print("Data is : ",t2)
```

```
#creating tuple using tuple( ) and tuple(iterable)
#tuple( ) -> tuple object
print("-"*30)
t3=tuple()
print("type is ",type(t3))
print("Data is : ",t3)

#tuple(iterable) # str | list | tuple | set | dict ....
print("-"*30)
lst=[10,20,30,40,50]
t4=tuple(lst)
print("Type is : ",type(t4))
print("Data is : ",t4)

#Example
print("-"*30)
t5=10,20,30,40,50
print("Type is : ",type(t5))
print("Data is : ",t5)

#Example Creating tuple objects with one element
print("-"*30)
t6=(10,)
print("Type is : ",type(t6))
```

```
print("Data is : ",t6)

#Creating a tuple with 1 to 10
#range(start,end,step) -> range object | iterable
print("-"*30)
r=range(1,11)
print("Typ is : ",type(r))
print("Data is : ",r)
#tuple(iterable) -> tuple
t7=tuple(r)
print("Type is : ",type(t7))
print("Data is : ",t7)

Reading the Values From Tuple Collections:
t1=(10,"A",3.14,None,120)
print("tuple : ",t1)

#Using Index
print("First : ",t1[0])
print("Last : ",t1[-1])

#Using Slicing
print("First 3 Objects : ", t1[0:3:1])
print("Last 3 Objects : ",t1[2:5])
```

```
#Unpacking
t2=(101,"ramesh",30,40,50)
no,name,m1,m2,m3=t2

print(no,name,m1,m2,m3,sep='-----')

#Unpacking using slicing
n,na=t2[0:2]
print("No is : ",n)
print("Name is : ",na)
marks=t2[2:5:1]
print("Marks is : ",marks)

#Reading all Objects From Tuple
import time
for i in t2:
    time.sleep(1)
    print(i)
```

Note:

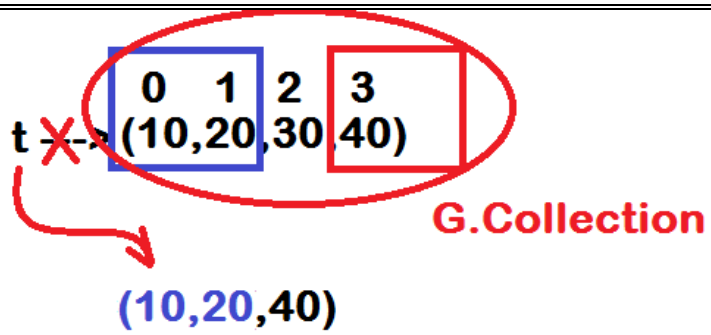
- **Where List Collection is mutable i.e.objects of list collection can be editable , But tuple is immutable**

collection , thus making the changes in the tuple is not possible

- append( ), insert( ), extend( ), pop( ), remove( ), clear(), copy( ) methods are not supported in tuple collection
- index( ) and count( ) methods are supported in tuple collections

```
t1=(10,"A",3.14,None,120,10)
print("tuple : ",t1)

#del t1[2]
#TypeError: 'tuple' object doesn't support item deletion

#t1[1]="AAA"
#TypeError: 'tuple' object does not support item assignment

#T.index(item[,start,[end] ]) -> int
pos=t1.index(10)
print("Object 10 found @ ",pos)
pos=t1.index(10,2,6)
print("Object 10 Found @ : ",pos)
```

```
#T.count(item) -> int
#T.count(item[,start,end]]) -> int [available int str]
no=t1.count(10)
print("Object 10 found for ",no," times ")

#Code for deleting an item from given position of tuple
collection
#     0   1   2   3  → index
t=(10,20,30,40)  #len(t) -> 4

print("Tuple is : ",t)
pos=int(input("enter pos of an object  ")) #pos=2
if pos<len(t):
   f=t[0:pos] #(10,20)
   s=t[pos+1:]
   t=f+s
   print("After Deleting an Object : ",t)
else:
   print("Sorry Invalid Index For deleting an Object")
```



Example code for inserting an Object @ specified position of Tuple :

```
#     0   1   2   3
t=(10,20,30,40)  #len(t) -> 4

print("Tuple is : ",t)
pos=int(input("enter pos of an object  ")) #pos=2

if pos<len(t):
   new=[input("Enter New Object ")  ] #str -> list
   f=t[0:pos]  #(10,20)
   s=t[pos:]  #(30,40)
   t=f+tuple(new)+s     #tuple(iterable) [new]list is iterable
   print("Result is : ",t)
else:
  print("Sorry Invalid pos")
```

App 2:

```
#     0   1   2   3
t=(10,20,30,40)  #len(t) -> 4

print("Tuple is : ",t)
pos=int(input("enter pos of an object  ")) #pos=2

if pos<len(t):
   new=int(input("Enter New Object ") )  #222
   f=t[0:pos]  #(10,20)
   s=t[pos:]  #(30,40)
   t=f+(new,)+s
   print("Result is : ",t)
else:
   print("Sorry Invalid pos")
```

Example For Updating an Existed Object with New @ specified Index Position:

```
#     0   1   2   3
t=(10,20,30,40)  #len(t) -> 4

print("Tuple is : ",t)
pos=int(input("enter pos of an object  ")) #pos=2

if pos<len(t):
   new=int(input("Enter new Object "))   #222
   f=t[0:pos]
   s=t[pos+1:]
   t=f+(new,)+s
   print("After Updating an Object : ",t)
else:
   print("Sorry Invalid Index ")
```

Note:

- List and Tuple is almost similar ,but list is mutable whereas tuple is immutable.
- List is Best if your frequent operations insertion and deletion
- Tuple is Best if your frequent operations are reading the values.

- **Tuple more faster than list collection as it is immutable the size fixed ,whereas list is mutable the size changing dynamically based on the values we inserted in to it.**