Python Data structure:- Tuple

In [2]:

```
#Tuple Data structure:-
#1.Order is applicable
#2.Duplicates are allowed
#3.Heterogeneous objects
#4.Indexing & slicing
#5.Immutable
#6.()
```

In [3]:

```
t=(1,2,3)
print(type(t))

```

```
<class 'tuple'>
```

In [5]:

```
t=1,2,3,4,5
print(t)
print(type(t))
```

```
(1, 2, 3, 4, 5)
<class 'tuple'>
```

Creation of tuple object:-

In [6]:

```
#1.empty tuple:-
t=()
print(type(t))
```

```
<class 'tuple'>
```

In [7]:

```
#2.Single valued tuple:-
t=10,
t1=(10,)
print(type(t))
print(type(t1))
```

```
<class 'tuple'>
<class 'tuple'>
```

In [8]:

```
#3.Multi value tuple:-
t4=(10,20,30)
t3=10,30,40
print(type(t4))
print(type(t3))
```

```
<class 'tuple'>
<class 'tuple'>
```

In [10]:

```
#by using tuple function:-
#t=tuple(sequence)
l=[10,20,30]
t=tuple(l)
print(t)
print(type(t))
t1=tuple("Akhil")
print(t1)
```

```
(10, 20, 30)
<class 'tuple'>
('A', 'k', 'h', 'i', 'l')
```

In [11]:

```
#with dynamic input:-
t=eval(input("enter tuple of value:"))
print(type(t))
```

```
enter tuple of value:(1,2,4,5)
<class 'tuple'>
```

In [12]:

```
#accesing tuple:-
t=(1,2,3,4)
print(t[0])
```

```
1
```

In [13]:

```
#slicing
t=(2,4,56,7)
print(t[0:2])
```

```
(2, 4)
```

Mathematical operators for tuple:-

In [14]:

```python
#1.Concatination operator(+)
#2.Repetition operator(*)
t1=(1,2,3,4,5)
t2=(6,7,8,9)
t3=t1+t2
print(t3)
#t4=t2+5======> TypeError
```

```
(1, 2, 3, 4, 5, 6, 7, 8, 9)
```

In [15]:

```python
t1=(1,2,3,4)
t2=t1*3
print(t2)
```

```
(1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4)
```

In [16]:

```python
t1=(3,4,6)
t2=(1,8,9,10)
t3=t1+t2
t4=t3*2
print(t3)
print(t4)
```

```
(3, 4, 6, 1, 8, 9, 10)
(3, 4, 6, 1, 8, 9, 10, 3, 4, 6, 1, 8, 9, 10)
```

In [19]:

```python
#Equality operators for tuple:-
#     ==,   !=
t1=("cat",'dog',"rat")
t2=("Cat","Dog","Rat")
t3=("CAT","DOG","RAT")
t4=("cat","dog","rat")
print(t1==t4)
print(t1!=t2)
print(t1==t3)
```

```
True
True
False
```

In [21]:

```
#Relational Operators:-
#     <,<=,>,>=
t1=(1,2,3)
t2=(4,5,6)
t3=(0,100)
t4=(100,)
print(t1<t4)
print(t2>=t1)
print(t3<=t4)
print(t3>=t1)
```

```
True
True
True
False
```

In [2]:

```
#Membership Operators
#    in, not in
t=(1,2,3,4,5,6)
print(3 in t)
print(100 in t)
print(3 not in t)
```

```
True
False
False
```

Important methods/function for tuple:-

In [3]:

```
#len()=====> returns no. of elements
t=(1,2,4,5,6,8,9,33)
print(len(t))

```

```
8
```

In [6]:

```
#count()=======>returns no. of occurancess of specified element
t=(4,3,1,5,6,1,4,6,4,8,0)
print(t.count(7))
```

```
0
```

In [7]:

```
#index()======>returns index of first occurences of specific element
t=(1,2,3,4,6,2,3,4)
print(t.index(2))
```

```
1
```

In [8]:

```
#reversing elements of tuple
t=(1,2,3,4,5)
r=reversed(t)
t1=tuple(r)
print(t1)
print(r)
```

```
(5, 4, 3, 2, 1)
<reversed object at 0x000001C1D7D21FD0>
```

In [9]:

```
#sorting of tuple:-
t=(5,14,57,8,1,4,77)
l=sorted(t)
print(l)
t1=tuple(l)
print(t1)
```

```
[1, 4, 5, 8, 14, 57, 77]
(1, 4, 5, 8, 14, 57, 77)
```

In [10]:

```
#max() & min() for tuple:-
t=(1,2,3,45,6,8,0)
print(min(t))
print(max(t))
```

```
0
45
```

In [11]:

```
#tuple packing and upacking
a=10
b=20
c=30
d=40
t=a,b,c,d
print(t)
```

```
(10, 20, 30, 40)
```

In [12]:

```
t=(1,2,3,4)
a,b,c,d=t
print("a=",a,"b=",b,"c=",c,"d=",d)
```

```
a= 1 b= 2 c= 3 d= 4
```

In [13]:

```
t=(20,30,40,50)
a,*b=t
print(a,b)
```

```
20 [30, 40, 50]
```

In [14]:

```
#program to take a tuple of numbers from the keyboard and print the sum and avg
t=eval(input("Enter a tuple with numbers:"))
sum=0
for i in t:
    sum=sum+i
print("The sum is",sum)
print("The avg is",sum/len(t))
```

```
Enter a tuple with numbers:(1,2,34,5,6)
The sum is 48
The avg is 9.6
```

In [ ]:

```

```