---:Tuple :---

In Python, tuples are immutables. Meaning, you cannot change items of a tuple once it is assigned. There are only two tuple methods count() and index() that a tuple object can call.

1. **Duplicates are allowed.**
2. **Order is preserved.**
3. **Objects are immutable.**
4. **Indexing is allowed.**
5. **Slicing is allowed.**
6. **Represented in parenthesis () with comma separated objects.**
7. **Homogeneous and Heterogeneous both objects are allowed.**

Tuple occupies less memory as compare to list, that's why tuple is more faster as compare to list**.**

Example:--

```
list = [10,20,30,40,50,60,70]
tuple = (10,20,30,40,50,60,70)
print(sys.getsizeof('Size of list = ',list))
print(sys.getsizeof('Size of tuple',tuple))

O/P-  64
      62
```

Built-in functions:-

1. **Len(tuple) # tuple variable must be a iterable.**
2. **Max(tuple)**
3. **Min(tuple)**
4. **Sum(tuple)**
5. **Tuple(list)**
6. **Type(tuple)**

Methods:--

1. **Count(obj). (How many occurrences)**

```
# Creating tuples
Tuple = (0, 1, (2, 3), (2, 3), 1, [3, 2],'Neeraj', (0), (0,))
res = Tuple.count((2, 3))
print('Count of (2, 3) in Tuple is:', res)

res = Tuple.count(0)
print('Count of 0 in Tuple is:', res)

res = Tuple.count((0,))
print('Count of (0,) in Tuple is:', res)

O/P:--
Count of (2, 3) in Tuple is: 2
Count of 0 in Tuple is: 2
Count of (0,) in Tuple is: 1
```

2. **Index(obj,start,stop)(obj is compulsory argument but rest are optional)**

```
Tuple = (0, 1, 2, 3, 2, 3, 1, 3, 2)
# getting the index of 3
res = Tuple.index(3)
print(res)
O/P:--
3
Tuple = (0, 1, 2, 3, 2, 3, 1, 3, 2)
# getting the index of 3
print(Tuple.index(3,4))
O/P:--
5
Tuple = (0, 1, 2, 3, 2, 3, 1, 3, 2)
# getting the index of 3
print(Tuple.index(3,0,4))
o/p:--
3
```

Tuple in Operators

1. Concatenation (+)

Combines two tuples.

```
a = (1, 2)
b = (3, 4)
print(a + b)

O/P:---
(1, 2, 3, 4)
```

2. Repetition (*)

Repeats a tuple a given number of times.

```
a = (1, 2)
print(a * 3)

O/P:---
(1, 2, 1, 2, 1, 2)
```

3. Membership (in, not in)

Checks if an element exists in the tuple.

```
t = (1, 2, 3)
print(2 in t)
print(5 not in t)

O/P:---
True
True
```

4. Indexing ([])

Accesses an element by index.

```
t = ('a', 'b', 'c')
print(t[1])
```

```
O/P:---
b
```

5. Slicing ([:])

Returns a portion of the tuple.

```
t = (0, 1, 2, 3, 4)
print(t[1:4])

O/P:---
(1, 2, 3)
```

6. Comparison Operators (==, !=, <, <=, >, >=)

Compares tuples lexicographically (element by element).

```
print((1, 2) == (1, 2))
print((1, 2) < (2, 0))
print((3, 4) >= (3, 2))

O/P:---
True
True
True
```