TUPLE

- Tuples are used to store multiple items in a single variable.
- We can store **heterogeneous** elements in a tuple.
- It **allows duplicate** elements.
- For accessing elements of tuple, we can use **positive** as well as **negative indexing**.
- Insertion **order** will be **preserved** in it.
- Tuple is **immutable.**
- The notation of tuple is **( )** brackets and the elements are separated by comma.
- In tuple, we can use **slicing** also.

Create Tuple with One Item:

To create a tuple with only one item, you have to add a comma after the item, otherwise Python will not recognize it as a tuple.

```
t = ("apple",)
print(type(t)) #<class 'tuple'>

#NOT a tuple
t = ("apple")
print(type(t)) #<class 'str'>
```

Syntax to create an empty tuple:

tuple_name = () or tuple_name = tuple()

Syntax to create tuple:

tuple_name = (ele1,ele2,ele3,…….)

or

tuple_name = ele1,ele2,ele3,….

e.g t = (1,3.5,'abc',True,1)

or

t = 1,3.5,'abc',True,1

Accessing tuple elements:

t = (10,20,30,40,50)

0 1 2 3 4 →+ve index (Left → Right)

-5 -4 -3 -2 -1 →-ve index (Right → Left)

t[0] = 10 , t[1] = 20
t[-1] = 50 , t[-2] = 40

Updating an element inside a tuple:

t = (10,20,30,40,50)
t[2] = 300
It will generate **ERROR** as tuple is **immutable.**

Iterating over the tuple:

```
t = (10,20,30,40,50)
for i in t:
    print(i)
```

OUTPUT:
10
20
30
40
50

Nested tuple:

It means a structure of multiple tuple inside tuple.

e.g

```
mh = ('Pune','Mumbai')
gj = ('Surat','Bhuj')

india = (mh , gj)
print(india)

for state in india:
    for cities in state:
        print(cities)
```

OUTPUT:

```
(('Pune','Mumbai') , ('Surat','Bhuj'))
Pune
Mumbai
Surat
Bhuj
```

Mathematical operations on tuple:

1. + (joining or merging of tuple)
   e.g t1 = (1,2,3)
   t2 = (4,5,6)
   t1+t2 #(1,2,3,4,5,6)

2. * (repetition of tuple)
   e.g t1 = (1,2,3)
   t1*3 #(1,2,3,1,2,3,1,2,3)

Methods in Tuple:

1.count – It counts the occurrence of the given element.
e.g. t = (10,20,10,40,10,60)
t.count(10) #3

2.index – It returns the index of first occurrence of given element.
e.g. t = (10,20,10,40,10,60)
t.index(10) #0

Common functions on tuple:

1.len(tuple) – It gives the total length of tuple.
2.max(tuple) – It gives maximum element from tuple.
3.min(tuple) – It gives minimum element from tuple.
4.sum(tuple) – It gives sum of all the tuple elements.
5.sorted(tuple) – This function returns a sorted version of the tuple. The sorting is in ascending order, and it doesn't modify the original tuple.

Python Tuples Packing:

You can also create a Python tuple without parentheses. This is called tuple packing.
e.g. t = 1, 4.5, 'xyz'

Python Tuples Unpacking:

Python tuple unpacking is when you assign values from a tuple to a sequence of variables in python.
e.g. percentages = (99,95,90,89,93,96)
a,b,c,d,e,f = percentages
print(c) #90

But here no. of elements should match with no. of variables to unpack.

Updating tuple elements:

To alter the contents of tuple we need to typecast it into any other data structure and after performing operations on that structure, we need to again re-typecast it into tuple.

e.g

```
t = (1,2,3)
l = list(t)
l.append(4)
l.extend([5,6,7])
t = tuple(l)
print(t) #(1,2,3,4,5,6,7)
```