Tuple is like list, it also store different hetrogenous values. But it is immutable and list is mutable

```
In [1]: #Create Tuple
        t=(1,2,3,4.5,"Ram")
```

```
In [2]: print(t)
```
```
(1, 2, 3, 4.5, 'Ram')
```

```
In [3]: type(t)
```
```
Out[3]: tuple
```

```
In [5]: #like lists, tuples also allow duplicates value lets see:
        list1=[1,2,2,4]
        tuple1=(3,4,4.5,4.5,"Ram")
```

```
In [6]: #to find length of tuple
        len(tuple1)
```
```
Out[6]: 5
```

```
In [7]: #will it work for list also?
        len(list1)
```
```
Out[7]: 4
```

```
In [8]: #yes length function work for both. As well as for string
        s='Hello'
        len(s)
```
```
Out[8]: 5
```

Access tuple item - Indexing and Slicing (Similar to lists and strings)

```
In [9]: print(t)
```
```
(1, 2, 3, 4.5, 'Ram')
```

```
In [10]: t[1]
```
```
Out[10]: 2
```

```
In [11]: t[-1]
```
```
Out[11]: 'Ram'
```

```
In [12]: t[0:3]
```
```
Out[12]: (1, 2, 3)
```

```
In [13]: t[-4:-1]
```
```
Out[13]: (2, 3, 4.5)
```

Change or add to tuple

```
In [14]: print(t)
```
```
(1, 2, 3, 4.5, 'Ram')
```

```
In [15]: t[4]="Shyam"
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 t[4]="Shyam"

TypeError: 'tuple' object does not support item assignment
```

```
In [16]: #What we get: Tuple does not support this assignment, lets look for list
         print(list1)
```
```
[1, 2, 2, 4]
```

```
In [17]: list1[1]=10
```

```
In [18]: print(list1)
```
```
[1, 10, 2, 4]
```

```
In [19]: #This is main difference between Tuple and Lists: immutable and mutable
```

```
In [19]:  #This is main difference between Tuple and Lists: Immutable and Mutable
```

```
In [20]:  #Lets see what count method does?
          print(tuple1)

          (3, 4, 4.5, 4.5, 'Ram')
```

```
In [21]:  #there are 4.5 two types
          tuple1.count(4.5)

Out[21]:  2
```

```
In [ ]:   #its showing there is 2s 4.5
```