Tuples

Tuples are the sequential datatype that stores multiple value in a single variable just like list. But there is one difference between the list and tuple is that we can make change in list but we cannot make change in tuple that's why tuple is unchangeable. But one thing python allow in tuple is that we can unpack the values of variable and extract it. Tuples are ordered because their each elements have index number. Tuples are indexed as index for each element exist. Tuples allow duplicate values. Tuple is always enclosed with small brackets. If you make a tuple of one element only then you have to add one comma for that in the tuple otherwise it throws error.

Accessing Elements in Tuple

To access elements of tuple you first have to write the name of the tuple after that you have to use opening square bracket and write the index number of the element you want to access and then you have to close the square bracket. To print the accessed element you have to put this **tuple[index]** inside the argument of the **print()** function.

In [11]:
```
#Accessing tuple elements
keshari = (1,4,7,0,7)
a = keshari[2]
print(a)
print(keshari[2])
```

```
7
7
```

Unpacking of Elements in Tuple

First of all, unpacking of tuple is a method in which we extract the values of variable back. Just like we have **t1 = (1,2)** then we assign each variable name for particular variable **ta,tb = t1**. It is noted that number of values present in tuple is equals to the number of variables. If it is not possible then use * asterick sign before the last variable as it takes all the values inside it.

In [13]:
```
#Unpack tuple elements
amrit = (23,56,88)
n1,n2,n3 = amrit
print(n1)
print(n2)
print(n3)
```

```
23
56
88
```

In [15]:
```
#Unpack tuple elements
keshari = (12,23,34,45,56,99)
n1,n2,*n3 = keshari
print(n1)
print(n2)
print(n3)
```

```
12
23
[34, 45, 56, 99]
```

Updating Elements in Tuple

To add element in tuple, first of all we have to convert the tuple into list. For converting tuple into list we have to use a constructor function for list that is **list()** function. It convert any datatype values into list. Then we can easily add elements in tuple. We can use **append()** function to add elements in tuple. To delete elements in tuple, you have to first convert the tuple into list using constructor function for list that is **list()** after that you can use **remove(value)** function to delete any element in tuple. Here, **remove(value)** is a builtin function for list to remove elements from list. It has one argument in which we pass the target value which we want to delete from the list. After passing any value in the argument of **remove()** function it matches the value from list if it find, it delete that value. We can also add tuple in tuple using addition operator (**+**).

In [20]:
```
#Add tuple elements
amritTuple = ("Hi","Hello","Fine","Hey")
print(amritTuple)
amritList = list(amritTuple)
amritList.append("Rang")
amritTuple = tuple(amritList)
print(amritTuple)
```

```
('Hi', 'Hello', 'Fine', 'Hey')
('Hi', 'Hello', 'Fine', 'Hey', 'Rang')
```

In [38]:
```
#Delete tuple elements
amritTuple = ("Lofi","Slowed","Music", "Slowed")
print(amritTuple)
amritList = list(amritTuple)
amritList.remove("Slowed")
amritTuple = tuple(amritList)
print(amritTuple)
```

```
('Lofi', 'Slowed', 'Music', 'Slowed')
('Lofi', 'Music', 'Slowed')
```

Printing Tuple Elements

To print the each elements of the tuple we have to iterate over each element of the tuple and it can be possible when we use looping statement. So, for printing the element of the tuple we use while loop, for-each loop and for loop.

Using while loop

In [26]:
```
amritTuple = (2,3,5,6,7)
i=0
while(i<len(amritTuple)):
    print(amritTuple[i])
    i = i + 1
```

```
2
3
5
6
7
```

Using for-each loop

In [30]:
```
amritTuple = (11,23,34,94)
for i in amritTuple:
    print(i)
```

```
11
23
34
94
```

Using for loop

In [33]:
```
amritTuple = (22,33,44,99)
for i in range(0,len(amritTuple),1):
    print(amritTuple[i])
```

```
22
33
44
99
```