# TUPLES

Tuples are used to store multiple items in a single variable. It is represented by parenthesis ( ) and specified by commas.

Tuples are Immutable(Can not be modified).

Tuples allow duplicates. Tuples are faster than lists.

Tuple makes the code safe from any accidental modification.

We can use a tuple to store the latitude and longitude of your home because a tuple always has a predefined number of elements(in the specific example, two). The same tuple type can be used to store the coordinates of other locations.

**Create a tuple:**

A tuple in Python is similar to a list but we can not change the elements of a tuple. A tuple can have any number of items and they may be of different types (integer,float,list,string,etc.)

```
Tuple=( )
print(Tuple)
O/P: <class 'tuple'>
```

**Access Tuple Element:** There ate various ways in which we can access the elements of a tuple.

Indexing: We can use the index operator to access an item in a tuple, where the index starts from 0. So, a tuple having 6 elements will have indices from 0-5.

```
Eg: Tuple=('S','U','N','N','Y')
    print(Tuple[0])
      O/P: 'S'
```

Negative Indexing: Python allows negative indexing for its sequences.

```
Eg: Tuple=('S','U','N','N','Y')
    print(Tuple[-1])
      O/P: 'Y'
```

**In-Built functions in Tuple:** There are few built in functions we call it as tuple operations.

**max():** To find the maximum value of a tuple.

```
Eg: T=(99,121,7,90,0.1)
    print(max(T))
```

O/P: 121

**Min():** To find the minimum value of a tuple.

**Eg:** T=(99,121,7,90,0.1)
print(min(T))
O/P: 0.1

**Sum():** To find the total sum of the elements in Tuple.

**Eg:** T=(99,121,7,90,0.1)
print(sum(T))
O/P: 317.1

**len():** To find the length of the tuple

**Eg:** T=(99,121,7,90,0.1)
print(len(T))
O/P: 5

**Tuple operations:**

**Concatenation:** Tuples can be concatenated by using the '+' Operation to create a new tuple element from both tuples.

**Eg:** t1=(1,2,3)
t2=(4,5,6)
concatenated_Tuple=t1+t2
O/P: (1,2,3,4,5,6)

**Repetition:** we can repeat a tuple multiple times using the '*' Operator

**Eg:** t1=(1,2,3)
t2=(4,5,6)
repeted_tuple=t1*3
O/P: (1,2,3,1,2,3,1,2,3)

**Slicing:** we can extract a subset of elements from an existing tuple

**Eg:** t3=('a','b','c','d','e')
New_tuple=t3[1:4]
O/P: ('b','c','d')

**Membership Operator:** We can check if an element is present in tuple or not it returns the Boolean value.

**Eg:** t3=('a','b','c','d','e')
'C' in t3
O/P:True

**Identity Operator:** we can check if both the tuples are same or not.

**Eg:** t1=(1,2,3)
t2=(4,5,6)

print(t1 is t2)  O/P: False  print(t1 is not t2)  O/P: True

**Iterating to a tuple:** we can use a for loop to iterate through each item in a tuple.

**Eg:** S=(1,22,3,44,5,66)

```
for i in S:
    print(i)
```

O/P: 1
22
3
44
5
66