

4.2 TUPLE

Tuples are sequence of values much like the list. The values stored in the tuple can be of any type and they are indexed by integers.

The main difference between list and tuple is **Tuple is immutable**. Tuple is represented using '()'

Syntax:

```
tuple_name=(items)
```

Ex:

```
>>> tuple1=('1','2','3','5')
>>> tuple2=('a','b','c')
>>> tuple3='3','apple','100'
```

TUPLES ARE IMMUTABLE:

The values of tuple cannot be changed.

```
>>> tuple1=('1','2','3','5')
tuple1[1]='4' #It Shows Error
```

Tuples can be immutable but if you want to add an item we can add it by

```
t1=('a','b')
t1=('A,')+t1[1:] → t1=('A','b')
```

The disadvantage in this method is we can only add the items from the beginning.

4.2.1 TUPLE ASSIGNMENT

Tuple Assignment means assigning a tuple value into another tuple.

Ex:

```
t=('Hello','hi')
>>> m,n=t
>>> print(m) → Hello
>>> print(n) → hi
>>> print(t) → Hello,hi
```

In order to interchange the values of the two tuples The following method is used.

```
>>>a=('1','4')
>>>b=('10','15')
>>>a,b=b,a
>>>print(a,b)
(('10','15'), ('1','4'))
```

COMPARING TUPLES

The comparison operator works with tuple and other sequence. It will check the elements of one tuple to another tuple. If they are equal it return true. If they are not equal it returns false.

```
>>>t1=('1','2','3','4','5')
>>>t2=('a','b','c')
>>>t1<t2 #It returns false
```

4.2.2 TUPLE AS RETURN VALUE

In a function a tuple can return multiple values where a normal function can return single value at a time.

Example-1:

Using a built-in function **divmod** which return quotient and remainder at the same time.

```
>>>t=divmod(7,3)
>>>print(t) → (2,1)
>>>quot,rem=divmod(7,3)
>>>print(quot) → 2
>>>print(rem) → 1
```

Example-2:

```
def swap(a,b,c):
    return(c,b,a)
a=100
```

```
b=200
```

```
c=300
```

```
>>>print("Before Swapping",a,b,c)
```

```
>>>print("After Swapping",swap(a,b,c))
```

Output:

Before Swapping 100,200,300

After Swapping 300,200,100

