In [ ]:
```
# Tuple

tuple is denoted by round brackets

- Tuple is Immutable(Can not modified after creation)

- Ordered :  the elements is  maintain their position

- Tuple is used the Indexed(0 - based)

- Heterogenous -->it can store different data types


# Characteristics

1- Tuple is Fatser than list for access operations

2- support mutiple assignment and Unpacking
```

In [2]:
```
x , y =  20, 10

def calc(x,y):
    return x, y

calc(10,20)
```

Out[2]: (10, 20)

In [4]:
```
# Operations

tup = ()
print(tup)
tup1 = (1,"john",2.5,True,[123,23,423,42,432,4])
print(tup1)
```

```
()
(1, 'john', 2.5, True, [123, 23, 423, 42, 432, 4])
```

In [9]:
```
tup2 = (1,)
type(tup2)
```

Out[9]: tuple

In [10]:
```
tup2 = ("1",)
print(type(tup2))
```

```
<class 'tuple'>
```

In [21]:
```
# unpacking
values = 10, 20,30,"hello world"
x,y,z,m = values  # unpacking
```

In [25]:
```
m
```

Out[25]: 'hello world'

In [ ]:
```
# Indexing and slicing
    --------------------> right to left (forword indexing)
```

```
    0,  1, 2,3
t = (10, 20,30,"hello world")

   left to right <---------- ( backword indexing)
    -1,-2,-3,-4
```

In [32]:
```
t = (10, 20,30,"hello world")

print(t[0])
print(t[3])
print(t[-2])

print(t[0:2])
```

```
10
hello world
30
(10, 20)
```

In [37]:
```
numbers = (2,4,5,6,7,8,9,2,2)

print(len(numbers))
print(max(numbers))
print(min(numbers))
print(numbers.count(2))
```

```
9
9
2
3
```

In [ ]:
```
a,b = 20,40

a,b  = b,a # swap values
```

In [ ]:
```

```