TUPLES

1. Tuple is similar to List except that the objects in tuple are immutable which means we cannot change the elements of a tuple once assigned.
2. When we do not want to change the data over time, tuple is a preferred data type.
3. Iterating over the elements of a tuple is faster compared to iterating over a list.

TUPLE CREATION

```python
In [2]: tup=() # Empty tuple
```

```python
In [4]: tup1=(10,20,30) # tuple of integer numbers
```

```python
In [6]: tup2=(4.7,5.0,6.9) # tuple of float numbers
```

```python
In [8]: tup3=('one','two','three') # tuple of strings
```

```python
In [10]: tup4=('Lahari',(1,2,3),(8,9,3)) # Nested tuples
```

```python
In [12]: tup5=(38,7.2,'One',True,) # mixed datatypes
```

```python
In [14]: tup6=('lahari',29,[56,78],[90,67],{'one','two'},(45,35))
```

```python
In [16]: len(tup6) # length of list
```

```
Out[16]: 6
```

Tuple Indexing

```python
In [19]: tup5 # whole list
```

```
Out[19]: (38, 7.2, 'One', True)
```

```python
In [21]: tup5[0] # retrieve first element of tuple
```

```
Out[21]: 38
```

```python
In [23]: tup5[3] # retrieve element at index location 3
```

```
Out[23]: True
```

```python
In [25]: tup3[1][0] # nested indexing - access the first character of the first  tuple el
```

```
Out[25]: 't'
```

```
In [27]: tup5[-1] # retrieve last element of the tuple
```

```
Out[27]: True
```

Tuple slicing

```
In [30]: tup5 # return whole tuple
```

```
Out[30]: (38, 7.2, 'One', True)
```

```
In [32]: tup5[0:3] # return all the elements from 0th to 3rd index location excluding the
```

```
Out[32]: (38, 7.2, 'One')
```

```
In [34]: tup5[2:6] # return all the elements from 0th to 6th index location excluding the
```

```
Out[34]: ('One', True)
```

```
In [36]: tup5[:2] # returns first two elements
```

```
Out[36]: (38, 7.2)
```

```
In [38]: tup5[2:] # returns the last two elements
```

```
Out[38]: ('One', True)
```

```
In [40]: tup5[-2:] # returns the last two elements
```

```
Out[40]: ('One', True)
```

```
In [42]: tup5[-1:] # retruns the last element of the tuple
```

```
Out[42]: (True,)
```

Remove & Change Elements

```
In [45]: del tup1[0] # Tuples are immutable which means can't delete the tuple elements
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[45], line 1
----> 1 del tup1[0]

TypeError: 'tuple' object doesn't support item deletion
```

```
In [ ]: tup1[0]=37 # Tuples are immutable which means can't change tuple elements
```

```
In [48]: del tup1 # deleting the whole tuple is possible
```

Loop through a tuple

In [51]:
```python
tup5
```

Out[51]: `(38, 7.2, 'One', True)`

In [53]:
```python
for i in tup5:
    print(i)
```

```
38
7.2
One
True
```

In [55]:
```python
for i in enumerate(tup5):
    print(i)
```

```
(0, 38)
(1, 7.2)
(2, 'One')
(3, True)
```

Tuple Membership

In [58]:
```python
tup5
```

Out[58]: `(38, 7.2, 'One', True)`

In [60]:
```python
'One' in tup5 # check if 'One' exist in the list
```

Out[60]: `True`

In [62]:
```python
'true' in tup5 # check if 'true' exist in the list
```

Out[62]: `False`

In [64]:
```python
if 'One' in tup5: # check if 'One' exist in the list
    print('One is present in the tuple')
```

```
One is present in the tuple
```

In [66]:
```python
if 'two' in tup5: # check if 'two' exists in the list
    print('two is present in the tuple')
else:
    print('two is not present in the tuple')
```

```
two is not present in the tuple
```

Index Position

In [69]:
```python
tup5
```

Out[69]: `(38, 7.2, 'One', True)`

In [71]:
```python
tup5.index(True) # Index of first element equal to True
```

Out[71]:   3

In [73]: `tup5.index(7.2) # Index of first element equal to 7.2`

Out[73]:   1

In [75]: `tup3`

Out[75]:   `('one', 'two', 'three')`

In [77]: `tup3.index('one') # Index of first element equal to 'one'`

Out[77]:   0

Sorting

In [80]: `tup2`

Out[80]:   `(4.7, 5.0, 6.9)`

In [82]: `sorted(tup2) # returns a new sorted list and doesn't change original tuple`

Out[82]:   `[4.7, 5.0, 6.9]`

In [84]: `sorted(tup2,reverse=True) # sort in descending order`

Out[84]:   `[6.9, 5.0, 4.7]`