TUPLE CREATION

In [4]:
```
tup1 = () # EMPTY TUPLE
```

In [6]:
```
tup2 = (10,30,60) # tuple of integers numbers
```

In [10]:
```
tup3 = (10.77,30.66,60.89) #TUPLE OF FLOAT NUMBERS
```

In [14]:
```
tup4 = ('one', 'two', "three") # tuple of strings
```

In [16]:
```
tup5 = ('Asif', 25 ,(50, 100),(150, 90)) # NESTED TUPLES
```

In [18]:
```
tup6 = (100, 'Asif', 17.765) # TUPLE OF MIXED DATA TYPES
```

In [20]:
```
tup7 = ('Asif', 25 ,[50, 100],[150, 90] , {'John' , 'David'} , (99,22,33))
```

In [22]:
```
len(tup7) # LENGTH OF LIST
```

Out[22]:
```
6
```

TUPLE INDEXING

In [25]:
```
tup2[0] # RETREIVE FIRST ELEMENT OF THE TUPLE
```

Out[25]:
```
10
```

In [27]:
```
tup4[0] # RETREIVE FIRST ELEMENT OF THE TUPLE
```

Out[27]:
```
'one'
```

In [29]:
```
tup4[0][0] # NESTED INDEXING - ACCESS THE FIRST CHARACTER OF THE FIRST TUPLE ELEMEN
```

Out[29]:
```
'o'
```

In [31]:
```
tup4[-1] # last item of the tuple
```

Out[31]:
```
'three'
```

In [33]:
```
tup5[-1] # last item of the tuple
```

Out[33]:
```
(150, 90)
```

TUPLE SLICING

In [36]:
```
mytuple = ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [38]:
```
mytuple[0:3] # RETURN ALL ITEMS FROM 0TH TO 3RD INDEX LOCATION EXCLUDING THE ITEM
```

Out[38]:
```
('one', 'two', 'three')
```

In [40]:
```
mytuple[2:5] # LIST ALL ITEMS FROM 2nd tO 5th INDEX LOCATION EXCLUDING THE ITEM
```

Out[40]:
```
('three', 'four', 'five')
```

In [42]:
```
mytuple[:3] # RETURN FIRST THREE ITEMS
```

Out[42]:
```
('one', 'two', 'three')
```

In [44]:
```
mytuple[:2] # RETURN FIRST TWO ITEMS
```

Out[44]:
```
('one', 'two')
```

In [46]:
```
mytuple[-3:] # RETURN LAST THREE ITEMS
```

Out[46]:
```
('six', 'seven', 'eight')
```

In [50]:
```
mytuple[-2:] # RETURN LAST TWO ITEMS
```

Out[50]:
```
('seven', 'eight')
```

In [52]:
```
mytuple[-1] # RETURN LAST ITEM OF THE TUPLE
```

Out[52]:
```
'eight'
```

In [54]:
```
mytuple[:] # RETURN WHOLE TUPLE
```

Out[54]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

REMOVE & CHANGE ITEMS

In [57]:
```
mytuple
```

Out[57]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [59]:
```
del mytuple[0] # TUPLES ARE IMMUTABLE WHICH MEANS WE CAN'T DELETE TUPLE ITEMS
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[59], line 1
----> 1 del mytuple[0]

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

In [63]:
```
mytuple[0] = 1 # TUPLES ARE IMMUTABLE WHICH MEANS WE CAN'T CHANGE TUPLE ITEMS
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[63], line 1
----> 1 mytuple[0] = 1

TypeError: 'tuple' object does not support item assignment
```

In [65]:
```
del mytuple # DELETING ENTIRE TUPLE OBJECT IS POSSIBLE
```

LOOP THROUGH A TUPLE

In [72]:
```
mytuple = ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [74]:
```
for i in mytuple:
    print(i)
```

```
one
two
three
four
five
six
seven
eight
```

In [76]:
```
for i in enumerate(mytuple):
    print(i)
```

```
(0, 'one')
(1, 'two')
(2, 'three')
(3, 'four')
(4, 'five')
(5, 'six')
(6, 'seven')
(7, 'eight')
```

TUPLE MEMBERSHIP

In [79]:
```
mytuple
```

Out[79]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [81]:
```
'one' in mytuple # CHECK IF 'ONE' EXIST IN THE LIST
```

Out[81]:
```
True
```

In [83]:
```
'ten' in mytuple # CHECK IF 'TEN' EXIST IN THE LIST
```

Out[83]:
```
False
```

In [85]:
```
if 'three' in mytuple: # CHECK IF 'THREE' EXIST IN THE LIST
    print('Three is present in the tuple')
else:
    print('Three is not present in the tuple')
```

```
Three is present in the tuple
```

In [87]:
```
if 'eleven' in mytuple: # CHECK IF 'ELEVEN' EXIST IN THE LIST
    print('eleven is present in the tuple')
else:
    print('eleven is not present in the tuple')
```

```
eleven is not present in the tuple
```

INDEX POSITION

In [90]:
```
mytuple
```

Out[90]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [92]:
```
mytuple.index('one') # INDEX OF FIRST ELEMENT EQUAL TO 'ONE'
```

Out[92]:
```
0
```

In [94]:
```
mytuple
```

Out[94]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [96]:
```
mytuple.index('one') # INDEX OF FIRST ELEMENT EQUAL TO 'ONE'
```

Out[96]:
```
0
```

In [98]:
```
mytuple.index('five') # INDEX OF FIRST ELEMENT EQUAL TO 'FIVE'
```

Out[98]:
```
4
```

In [106…
```
mytuple1 = ('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

In [108…
```
mytuple1
```

Out[108…
```
('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

In [110…
```
mytuple1.index('one') # INDEX OF FIRST ELEMENT EQUAL TO 'ONE'
```

Out[110…
```
0
```

SORTING

In [113…
```
mytuple2 = (43,67,99,12,6,90,67)
```

In [115…
```
sorted(mytuple2) # RETURNS A NEW SORTED LIST AND DOESN'T CHANGE ORIGINAL TUPLE
```

Out[115…
```
[6, 12, 43, 67, 67, 90, 99]
```

In [117…
```
sorted(mytuple2, reverse=True) # SORT IN DESCENDING ORDER
```

Out[117…
```
[99, 90, 67, 67, 43, 12, 6]
```

SETS

- 1. Unordered & Unindexed collection of items.
- 2. Set elements are unique. Duplicate elements are not allowed.
- 3. Set elements are immutable (cannot be changed).
- 4. Set itself is mutable. We can add or remove items from it.

SET CREATION

In [1]:
```
myset = {1,2,3,4,5} # Set of numbers
myset
```

Out[1]:
```
{1, 2, 3, 4, 5}
```

In [2]:
```
len(myset) #Length of the set
```

Out[2]:
```
5
```

In [3]:
```
my_set = {1,1,2,2,3,4,5,5} # Dup.icate elements are not allowed.
my_set
```

Out[3]:
```
{1, 2, 3, 4, 5}
```

In [4]:
```
myset1 = {1.79,2.08,3.99,4.56,5.45} # Set of float numbers
myset1
```

Out[4]:
```
{1.79, 2.08, 3.99, 4.56, 5.45}
```

In [5]:
```
myset2 = {'Asif' , 'John' , 'Tyrion'} # Set of Strings
myset2
```

Out[5]:
```
{'Asif', 'John', 'Tyrion'}
```

In [ ]:
```

```