



TUPLE

TUPLE creation

```
In [6]: t1 = () # empty tuple  
t1
```

```
Out[6]: ()
```

```
In [4]: t2 = (10,20,30) # tuple of integer numbers  
t2
```

```
Out[4]: (10, 20, 30)
```

```
In [7]: t3 = (10.77,30.66,60.89) # tuple of float numbers  
t3
```

```
Out[7]: (10.77, 30.66, 60.89)
```

```
In [8]: t4 = ('one','two','three') # tuple of string  
t4
```

```
Out[8]: ('one', 'two', 'three')
```

```
In [9]: t5 = ('savi',25,(50,100),(150,90)) # nested tuples  
t5
```

```
Out[9]: ('savi', 25, (50, 100), (150, 90))
```

```
In [10]: t6 = (100,'savi',17.765) # tuple of mixed data types  
t6
```

```
Out[10]: (100, 'savi', 17.765)
```

```
In [11]: t7 = ('savi',25,[50,100],[150,90],{'dev','prakash sir'},(99,22,33))  
t7
```

```
Out[11]: ('savi', 25, [50, 100], [150, 90], {'dev', 'prakash sir'}, (99, 22, 33))
```

```
In [12]: len(t7)
```

```
Out[12]: 6
```

TUPLE indexing

```
In [14]: t2[0] # retrieve first element of the list
```

```
Out[14]: 10
```

```
In [15]: t4[0] # retrieve first element of the tuple
```

```
Out[15]: 'one'
```

```
In [16]: t4[0][0] # Nested indexing - access the first character of the first tuple element
```

```
Out[16]: 'o'
```

```
In [17]: t4[-1] # last element of the tuple
```

```
Out[17]: 'three'
```

```
In [18]: t2
```

```
Out[18]: (10, 20, 30)
```

```
In [19]: t5
```

```
Out[19]: ('savi', 25, (50, 100), (150, 90))
```

```
In [21]: t5[-1] # last element of the tuple
```

```
Out[21]: (150, 90)
```

tuple SLICING

```
In [39]: myt = ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')  
myt
```

```
Out[39]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [23]: myt[0:3] # return all items from 0th TO 3rd index location
```

```
Out[23]: ('one', 'two', 'three')
```

```
In [24]: myt[2:5] # list of all items from 2nd to 5th index location
```

```
Out[24]: ('three', 'four', 'five')
```

```
In [25]: myt[:3] # return first three items
```

```
Out[25]: ('one', 'two', 'three')
```

```
In [26]: myt[:2]    #return frst two items
```

```
Out[26]: ('one', 'two')
```

```
In [40]: myt[-3:]   #return last three items
```

```
Out[40]: ('six', 'seven', 'eight')
```

```
In [41]: myt[-2:]   #return last two items
```

```
Out[41]: ('seven', 'eight')
```

```
In [42]: myt[-1]
```

```
Out[42]: 'eight'
```

```
In [43]: myt[:]     # return whole tuple
```

```
Out[43]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

REMOVE & CHANGE items

```
In [44]: del myt[0]   # Tuples are immutable which means we can't DELETE Tuple items
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[44], line 1  
----> 1 del myt[0]  
  
TypeError: 'tuple' object doesn't support item deletion
```

```
In [45]: myt[0] = 1   # Tuples are immutable which means we can't CHANGE Tuple items
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[45], line 1  
----> 1 myt[0] = 1  
  
TypeError: 'tuple' object does not support item assignment
```

```
In [46]: del myt     # DELETING entire TUPLE OBJECT is possible
```

LOOP through a TUPLE

```
In [48]: myt = ('one','two','three','four','five','six','seven','eight')
```

```
myt
```

```
Out[48]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [49]: for i in myt:  
         print(i)
```

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

```
In [50]: for i in enumerate(myt):  
         print(i)
```

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

COUNT

```
In [51]: myt1 = ('one','two','three','four','one','one','two','three')
```

```
In [53]: myt1.count('one') # no. of times item 'one' occurred in tuple
```

```
Out[53]: 3
```

```
In [54]: myt1.count('two')
```

```
Out[54]: 2
```

```
In [55]: myt1.count('four')
```

```
Out[55]: 1
```

TUPLE membership

```
In [56]: myt
```

```
Out[56]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [57]: myt1
```

```
Out[57]: ('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

```
In [58]: 'one' in myt
```

```
Out[58]: True
```

```
In [59]: 'ten' in myt
```

```
Out[59]: False
```

```
In [61]: if 'three' in myt:
          print(" three is present in the tuple")  # chek if 'three' is exist in th
        else:
          print('three is not present in the tuple')
```

```
three is present in the tuple
```

```
In [62]: if 'eleven' in myt:
          print(" eleven is present in the tuple")  # chek if 'three' is exist in t
        else:
          print('eleven is not present in the tuple')
```

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

INDEX position

```
In [63]: myt.index('one')  # INDEX of frst element equal to 'one'
```

```
Out[63]: 0
```

```
In [65]: myt.index('five')
```

```
Out[65]: 4
```

```
In [66]: myt1
```

```
Out[66]: ('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

```
In [67]: myt.index('one')
```

```
Out[67]: 0
```

```
In [68]: myt.index('two')
```

```
Out[68]: 1
```

sorting

```
In [69]: myt2 = (43,67,99,12,6,90,67)
         myt2
```

```
Out[69]: (43, 67, 99, 12, 6, 90, 67)
```

```
In [71]: sorted(myt2)    # return a new list and doesn't change original tuple
```

```
Out[71]: [6, 12, 43, 67, 67, 90, 99]
```

```
In [72]: sorted(myt2, reverse = True)
```

```
Out[72]: [99, 90, 67, 67, 43, 12, 6]
```

```
In [ ]:
```

```
In [ ]:
```