Tuple

In Python, tuple is an immutable Tuple Syntax: A tuple is created by using parentheses()

Tuple Creation

In [13]:

```
tup1=() # Empty tuple
tup1
```

Out[13]:

```
()
```

In [15]:

```
print(type(tup1))
```

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

In [17]:

```
tup2=(10,30,60) # Tuple of integers numbers
tup2
```

Out[17]:

```
(10, 30, 60)
```

In [19]:

```
tup3=(10.77,30.66,69.89) #Tuple of float numbers
tup3
```

Out[19]:

```
(10.77, 30.66, 69.89)
```

In [21]:

```
tup4=('one','two','three') # Tuple of strings
tup4
```

Out[21]:

```
('one', 'two', 'three')
```

In [23]:

```
tup5=('sri',25,(50,100),(150,90)) #Nested Tuples
tup5
```

Out[23]:

```
('sri', 25, (50, 100), (150, 90))
```

In [27]:

```
tup6=(100,'sri',18.09,True) #Tuple of mixed data types
tup6
```

Out[27]:

```
(100, 'sri', 18.09, True)
```

In [34]:

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

Out[34]:

```
('Sri', 25, [50, 100], [150, 90], {'David', 'John'}, (99, 22, 33))
```

In [38]:

```
len(tup7) # Length of string
```

Out[38]:

```
6
```

Tuple Indexing

```
In [41]: tup2
```

```
Out[41]: (10, 30, 60)
```

```
In [43]: tup2[0] # Retriving 1st element of tuple
```

```
Out[43]: 10
```

```
In [45]: tup2[-1] # Retriving last element of tuple
```

```
Out[45]: 60
```

```
In [47]: tup4[-1] # Last item of tuple
```

```
Out[47]: 'three'
```

```
In [49]: tup4[0][0] # Nested Indexing - Access the first character of the first tuple elemen
```

```
Out[49]: 'o'
```

```
In [51]: tup5[-1] # Last item of tuple
```

```
Out[51]: (150, 90)
```

Tuple Slicing

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

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

```
In [58]: mytup[0:3] # Return all items from 0th to 3rd index location excluding the item at
```

```
Out[58]: ('one', 'two', 'three')
```

```
In [60]: mytup[2:5] # All items from 2nd to 5th index location excluding the item at 5th ind
```

```
Out[60]: ('three', 'four', 'five')
```

```
In [62]: mytup[:3] # Return first 3 items
```

```
Out[62]: ('one', 'two', 'three')
```

```
In [64]: mytup[:2] # Return first 2 items
```

```
Out[64]: ('one', 'two')
```

```
In [66]: mytup[-3:] # Return last 3 items
```

```
Out[66]: ('six', 'seven', 'eight')
```

```
In [68]: mytup[-2:] # Return last 2 items
```

```
Out[68]: ('seven', 'eight')
```

```
In [70]: mytup[-1] # Return last item of tuple
```

```
Out[70]: 'eight'
```

```
In [74]: mytup[:] #Return whole tuple
```

```
Out[74]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

Remove and Change Items

```
In [77]: mytup
```

```
Out[77]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [79]: del mytup[0] # Tuples are immutable which means we can't DELETE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[79], line 1
----> 1 del mytup[0]

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

```
In [81]: mytup[0]=1 # Tuples are immutable which means we can't CHANGE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[81], line 1
----> 1 mytup[0]=1

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

```
In [89]: del mytup # Deleting entire tuple object is possible
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[89], line 1
----> 1 del mytup

NameError: name 'mytup' is not defined
```

Loop through a tuple

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

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

```
In [96]: for i in mytup:
             print(i)

one
two
three
four
five
six
seven
eight
```

```
In [98]: for i in enumerate (mytup):
             print(i)

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

Tuple Membership

```
In [101… mytup

Out[101… ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [103… 'one' in mytup #Check if'one' exist in the list

Out[103… True
```

```
In [105… 'eight' in mytup # Check if 'eight' exist in the list

Out[105… True
```

```
In [107… 'ten' in mytup # Check if 'ten' exist in the list

Out[107… False
```

```
In [109… if 'three' in mytup: # Check if 'three' is exist in the list
             print('Three is present in mytup')
         else:
             print('Three is not present in mytup')

Three is present in mytup
```

In [111…

```
if 'eleven' in mytup: # Check if 'eleven' is exist in the list
    print('Eleven is present in mytup')
else:
    print('Eleven is not present in mytup')
```

```
Eleven is not present in mytup
```

Index Position

In [114…

```
mytup
```

Out[114…

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [116…

```
mytup.index('one') # Index of first element equal to 'one'
```

Out[116…

```
0
```

In [118…

```
mytup.index('five') # Index of first element equal to 'five'
```

Out[118…

```
4
```

In [122…

```
mytup1 =('one','two','three','four','one','one','two','three')
mytup1
```

Out[122…

```
('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

In [124…

```
mytup1.index('one') # Index of first element equal to'one'
```

Out[124…

```
0
```

Sorting

In [129…

```
mytup2=(43,67,99,27,9,80,77)
mytup2
```

Out[129…

```
(43, 67, 99, 27, 9, 80, 77)
```

In [131…

```
sorted(mytup2) # Returns a new sorted list and doesn't change original tuple
```

Out[131…

```
[9, 27, 43, 67, 77, 80, 99]
```

In [133…

```
sorted(mytup2,reverse=True) # Sort in Descending Order
```

Out[133…

```
[99, 80, 77, 67, 43, 27, 9]
```