Day6_Tuple_Built-in_DS_2

May 23, 2025

1 Day 6: Tuples in Python

Today I learned about **tuples** in Python. A tuple is a collection used to store multiple items in a single variable. Unlike lists, tuples are **immutable**, which means once created, their values cannot be changed.

You create a tuple by placing items inside **parentheses** like this: `t = (1, 'Two', 3.0)`

Tuples are useful when you want to make sure the data stays unchanged. For example, **bank records** or fixed settings can be stored as tuples.

Although tuples themselves can't be modified, they can contain **mutable objects** like lists. If a list is inside a tuple, you can still change the list's contents:

- Example: `t = (1, ['a', 'b', 'd'], 0)` Change: `t[1][2] = 'c'` Result: `(1, ['a', 'b', 'c'], 0)` So yes, **it's okay to change the list inside a tuple**.

Tuples support only two main functions:

- `index()` – tells you the position of an item
- `count()` – counts how many times an item appears

Functions like `append()`, `remove()`, or `sort()` won't work on tuples, because they are designed for lists which are changeable.

Tuples can be:

- Nested (you can have a tuple inside a tuple)
- Contain lists (you can have a list inside a tuple)

An **empty tuple** is written like this: `()`

Also, tuples (like strings and lists) are **iterable and ordered**. This means their items can be looped through, and they have a fixed order.

You can also create a tuple using the **tuple() constructor**.

2 Tuple creation

[2]:
```
tup1 = () # Empty tuple
tup2 = (10,20,30,60) # tuple of integers numbers
tup3 = (10.77,30.66,60.89) # tuple of flots numbers
tup4 = ("one",'two','Three') # tuple of strings
```

```
tup5 = ("Akshay",25,(50,100),(150,90)) #nested tuples
tup6 = (100,'Akki',17.765) #tuple of mix data types
tup7 = ('Akshay',25,[50,100],[150,90],{'Jonny','Sins'},(99,22,33))
len(tup7) #Length of list
```

[2]: 6

[3]:
```
print('tup1 :',tup1)
print('tup2 :',tup2)
print('tup3 :',tup3)
print('tup4 :',tup4)
print('tup5 :',tup5)
print('tup6 :',tup6)
print('tup7 :',tup7)
```

```
tup1 : ()
tup2 : (10, 20, 30, 60)
tup3 : (10.77, 30.66, 60.89)
tup4 : ('one', 'two', 'Three')
tup5 : ('Akshay', 25, (50, 100), (150, 90))
tup6 : (100, 'Akki', 17.765)
tup7 : ('Akshay', 25, [50, 100], [150, 90], {'Sins', 'Jonny'}, (99, 22, 33))
```

3 Tuple Indexing

[4]:
```
tup2[0] # Retreive first element of the tuple
```

[4]: 10

[5]:
```
tup4[0] # Retreive first element of the tuple
```

[5]: 'one'

[6]:
```
tup4[0][0] #Nested Indexing - Access the frist character of the first tuple␣
↪element
```

[6]: 'o'

[7]:
```
tup4[-1] # Last item of the tuple
```

[7]: 'Three'

4 Tuple Slicing

[8]:
```
mytuple = ('one','two','three','four','five','six','seven','eight')
```

[9]:
```
mytuple[0:3] #Return all items from 0th to 3rd index location
```

[9]:
```
('one', 'two', 'three')
```

[10]:
```
mytuple[2:5] #Return all items from 2th to 5th index location
```

[10]:
```
('three', 'four', 'five')
```

[11]:
```
mytuple[:3] #Return first three items
```

[11]:
```
('one', 'two', 'three')
```

[12]:
```
mytuple[:2] #Return first 2 items
```

[12]:
```
('one', 'two')
```

[13]:
```
mytuple[-3:] #Return last three items
```

[13]:
```
('six', 'seven', 'eight')
```

[14]:
```
mytuple[-2:] #Return last 2 items
```

[14]:
```
('seven', 'eight')
```

[15]:
```
mytuple[-1] #Return last item of tuple
```

[15]:
```
'eight'
```

[16]:
```
mytuple[:] # Return whole tuple
```

[16]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

5 Remove & change items

[17]:
```
mytuple
```

[17]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

[18]:
```
del mytuple[0] # Tuples are immutable which means we can not DELETE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 del mytuple[0]

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

[19]:
```
mytuple[0] = 1 # Tuples are immutable which means we can not CHANGE tuple␣
↪items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[19], line 1
----> 1 mytuple[0] = 1

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

[ ]:
```
del mytuple # Deleting entire tuple object is possible
```

6 Loop Through a Tuple

[20]:
```
mytuple
```

[20]:
```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

[21]:
```
for i in mytuple:
    print(i)
```

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

[22]:
```
for i in enumerate(mytuple):
    print(i)
```

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

7 Tuple Membership

[23]:
```
mytuple
'one' in mytuple # Check of 'one' exist in the tuple
```

[23]:
```
True
```

[24]:
```
'ten' in mytuple # Check of 'ten' exist in the tuple
```

[24]: False

[25]:
```
if 'three' in mytuple:    # Check if 'three' exist in the List
    print('Yes, Three is present in the Tuple')
else:
     print('No, Three is not present in the Tuple')
```

Yes, Three is present in the Tuple

[26]:
```
if 'Twenty' in mytuple:    # Check if 'Twenty' exist in the List
    print('Yes, Twenty is present in the Tuple')
else:
     print('No, Twenty is not present in the Tuple')
```

No, Twenty is not present in the Tuple

8 Index Position

[27]:
```
mytuple
```

[27]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')

[28]:
```
mytuple.index('one') #Index of first elemnts equal to 'one'
```

[28]: 0

[29]:
```
mytuple.index('five') #Index of first elemnts equal to 'five'
```

[29]: 4

9 Sorting

[30]:
```
mytuple1 = (1,4,5,7,33,12,34,76,89,90,1,2)
```

[31]:
```
mytuple1
```

[31]: (1, 4, 5, 7, 33, 12, 34, 76, 89, 90, 1, 2)

[32]:
```
sorted(mytuple1) # Retuen a new sorted list and does not change original tuple
```

[32]: [1, 1, 2, 4, 5, 7, 12, 33, 34, 76, 89, 90]

[33]:
```
sorted(mytuple1, reverse =True) #Sort in descending order
```

[33]: [90, 89, 76, 34, 33, 12, 7, 5, 4, 2, 1, 1]