

Tuple

Tuple is one of the built in data type in python. Tuple literals are written within square brackets () where elements are separated by comma.

Characteristics of a Python Tuple

The various characteristics of a tuple are:

- Ordered: Lists maintain the order in which the data is inserted.
- immutable: In list element(s) are **not** changeable. This is one of the difference between the list and tuple.
- Heterogenous: Tuple can store elements of various data types.
- Duplicate Elements: Tuple allow us to store duplicate data.

```
In [2]: # We can use round bracket(parentheses) without any element to create the empty tuple
t1 = ()
print(t1)
```

()

```
In [4]: # Enter the elements and separate by comma inside the round bracket to create the tuple
t=(1,2,3,4,5,6)      #Integer List
print(t)
```

(1, 2, 3, 4, 5, 6)

```
In [5]: t=(1.5, 2.5, 6.2, 8.2, 15.6)      #Float List
print(t)
```

(1.5, 2.5, 6.2, 8.2, 15.6)

```
In [6]: t=('A', "B", "C", 'D')      #String List
print(t)
```

('A', 'B', 'C', 'D')

```
In [7]: t=(10, 35, 45.62, 78.12, "Hello", 'Abhi')      #Heterogeneous datatype list
print(t)
```

(10, 35, 45.62, 78.12, 'Hello', 'Abhi')

```
In [8]: t=(10, 35, 20, 65, 35, 10, 10)      #Duplicate items in the list
print(t)
```

(10, 35, 20, 65, 35, 10, 10)

```
In [9]: t=(10, 35, 45.62, 78.12, ("Hello", 'Abhi'), 12)      #Nested list
print(t)
```

(10, 35, 45.62, 78.12, ('Hello', 'Abhi'), 12)

```
In [10]: t=(10, 35, 45.62, 78.12, ["Hello", 'Abhi'], 12)      #Nested list in tuple
print(t)
```

(10, 35, 45.62, 78.12, ['Hello', 'Abhi'], 12)

```
In [19]: t1=10, 20, 30
print(f'The value of t1 is {t1}.')
```

The value of t1 is (10, 20, 30).

```
In [20]: x,y,z=t1
print(f"The value of x is {x}, the value of y is {y} and the value of z is {z}.")
```

The value of x is 10, the value of y is 20 and the value of z is 30.


```
In [1]: t=tuple((1,2,3,4,5))    # The tuple() function creates a tuple object.
print(t)
```

(1, 2, 3, 4, 5)

```
In [22]: print("Length of tuple", len(t1))    # len() function
print("Sum of tuple", sum(t1))                # sum() function
print("Max value in tuple", max(t1))          # max() function
print("Min value in tuple", min(t1))          # min() function
```

Length of tuple 3
Sum of tuple 60
Max value in tuple 30
Min value in tuple 10

Note: Since tuple is immutable which means we cannot modify the tuple like list that is we can not either add elements in the tuple or cannot remove elements from the tuple.

We can fetch the elements from the tuple in the same way as we do in List that is we can use indexing and slicing here in the tuple. To know about indexing and slicing click on this link  <https://github.com/Abhishekk-Git/List-in-Python/blob/main/List-Python.ipynb>

Tuple have only two methods.

- index()
- count()

index() method

The index() method gives the index of a particular element from the tuple.

Syntax: Tuple.index(element, start, end)

The index() method can take a maximum of three arguments:

- element - the element to be searched.
- start (optional) - start searching from this index
- end (optional) - search the element up to this index

The index() method returns the index of the given element in the tuple. If the element is not found, a ValueError exception is raised.

```
In [11]: t=["Python", "Java", "HTML", "JavaScript"]
print(t.index("HTML"))    #--> Here, the index of element "HTML" is returned.
```

2

```
In [12]: t=["Python", "Java", "HTML", "JavaScript"]
print(t.index("Java"))    #--> Here, the index of element "HTML" is returned.
```

1

Note: The index() method only returns the first occurrence of the matching element.

```
In [13]: l = [ 2, 5, 6, 5, 7, 5, 8, 9]
l.index(5, 3, 8)
# Here 5 is the element that we need to look for, 3 is the start index and 8 is the end index
# 5 within the range of index 3 to 8.

Out[13]: 3
```

count() method

The count() method returns the number of times the specified element appears in the tuple.

Syntax: Tuple.count(element)

The count() method can take one argument and return the number of times that element appear in the tuple.

```
In [14]: t = (2, 5, 6, 5, 7, 5, 8, 9)
t.count(5)
# Here 5 is the element that we need to look for. 5 comes three times in the tuple

Out[14]: 3
```

Advantages of Tuple over List in Python

Since tuples are quite similar to lists, both of them are used in similar situations.

- We generally use tuples for heterogeneous (different) data types and lists for homogeneous (similar) data types.
- Since tuples are immutable, iterating through a tuple is faster than with a list. So there is a slight performance boost.
- Tuples that contain immutable elements can be used as a key for a dictionary. With lists, this is not possible.
- If you have data that doesn't change, implementing it as tuple will guarantee that it remains write-protected.

Click this link to learn more:  <https://github.com/Abhishekk-Git>