
TUPLE IN PYTHON:

- Tuple is a sequence data type.
- Tuple is a heterogamous collection of data.
- Tuple is an immutable data type i.e. it cannot be changed after being created.
- Tuple is denoted by parenthesis i.e. ( ) and the elements of the tuple are comma separated.

How to create a tuple in Python:

#Create a tuple with three integer element

```
>>> T= (8, 6, 7)
>>> T
(8, 6, 7)
>>> type(T)
<class 'tuple'>
```

#Create a tuple with a single element. Comma must be used after the element.

```
>>> T= (5,)
>>> T
(5,)
>>> type(T)
<class 'tuple'>
```

#Without parenthesis comma separated values are also treated as tuples.

```
>>> T=5,
>>> T
(5,)
```

#Without a comma, a single element is not tuple.

```
>>> T = (5)
>>> T
5
>>> type(T)
<class 'int'>
```

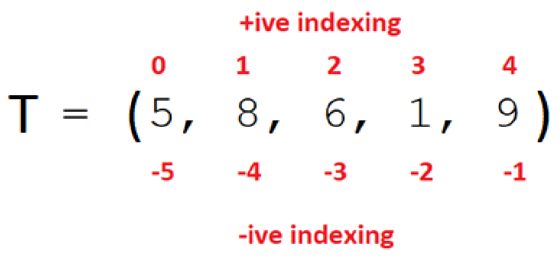
#Following is string, not tuple

```
>>> T= ('Hello')
>>> T
'Hello'
>>> type(T)
<class 'str'>
>>> T= (5, 9.8,'Hello', 9)
>>> T
(5, 9.8, 'Hello', 9)
```

#Create a tuple without parentheses.

```
>>> T=1, 2, 3, 4
>>> T
(1, 2, 3, 4)
>>> type(T)
<class 'tuple'>
```

Tuple Indexing



```
>>> T= (5, 8, 6, 1, 9)
>>> T
(5, 8, 6, 1, 9)
>>> T[2]
6
>>> T[-3]
6
>>> T[6]
IndexError: tuple index out of range
```

Tuple Operations

Concatenation

Using + operator

```
>>> T1=(1,2,3)
>>> T2=(4,5,6)
>>> T=T1+T2
>>> T
(1, 2, 3, 4, 5, 6)
```

Repetition

Tuple can be created using the repetition operator, *.
The repetition operator makes multiple copies of a tuple and joins them all together. The general format is: T * n,
Where T is a tuple and n is the number of copies to make.

```
>>> T= (1, 2)*3
>>> T
(1, 2, 1, 2, 1, 2)
```

In this example, (1, 2) is a tuple with two elements 1 and 2, The repetition operator makes 3 copies of this tuple and joins them all together into a single tuple.

Membership

Membership operator is used to check or validate the membership of an element in the tuple or sequence.
There are two types of membership operators
1. in operator: It return True if a element with the specified value is present in the tuple
2. not in operator: It returns True if an element with the specified value is not present in the tuple.
Examples:

```
T = (5,"India", 8, 6, 1)
>>> 5 in T
   True
>>> "India" in T
   True
>>> 9 in T
   False
>>> 9 not in T
   True
```

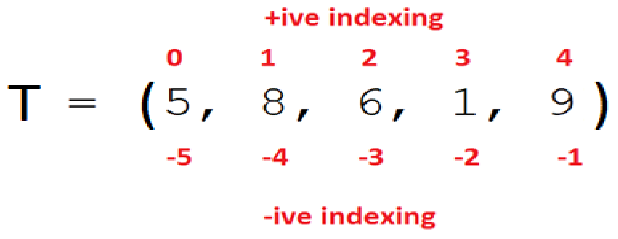
Slicing in Tuple:

Note: **Slicing never give IndexError**

Syntax:

Object_name[start : stop : interval]

- Here start, stop and interval all have default values.
- Default value of start is 0, stop is n (if interval is +ive) and -1 if interval is –ive
- Default value interval is 1
- In slicing start is included and stop is excluded.
- If interval is +ive then slicing is performed from left to right. For Ex:



```
>>> T[1:4:1]
(8, 6, 1)
```

If we give an index out of range then it is considered as the default value.

```
>>> T[1:5:2]
(8, 1)
```

If start index is not given then it consider as default value

```
>>> T[:5:2]
(5, 6, 9)
```

If start>=stop and interval is positive the slicing will return empty list

```
>>> T[4:2]
()
```

If interval is negative slicing performed from right to left

Access the list element where start index is 4 and stop index is 1 in reverse order

```
>>>T[4:1:-1]
(9, 1, 6)
```

Access the list element where start index is 5 and stop index is 0 with interval 2 in reverse order

```
>>>T[5:0:-2]
(9, 6)
```

```
>>>T[-1:-4:-1]
(9, 1, 6)
>>>T[-1:-4:-1]
(9, 1, 6)
```

Built-in functions/methods of tuple:

1: len():

len() is a function of tuple.
It returns the total number of elements in the tuple i.e. length of the tuple.
For ex:

```
>>> T = (8, 6, 'Hello', 9, 1)
>>> len(T)
5
```

It returns the total numbers of elements in the tuple i.e. 5

2: tuple()

tuple() is a function used to convert an object into tuple.
For Ex:

```
>>> S="Python"
>>> T=tuple(S)
>>> T
('P', 'y', 't', 'h', 'o', 'n')
```

In the above example, a string S is converted into the tuple.

3: count()

count() is a function used to count() the number of occurrences of an element in the tuple.

```
>>> T = (1, 2, 3, 2, 4, 2, 5)
>>> T.count(2)
3
```

4: index()

It returns the index of the first occurrence of the element

```
>>> T = (1, 2, 3, 2, 4, 2, 5)
>>> T.index(2)
1
```

5: sorted()

It is a built-in function to sort the elements of Tuple.

It return a new list of sorted elements of the tuple in ascending order (by default)
It does not change the existing tuple.

```
T = (1, 2, 3, 2, 4, 2, 5)
print(sorted(T))
Output:
[1, 2, 2, 2, 3, 4, 5]
```

For sort the elements in descending order use reverse = True

```
T = (1, 2, 3, 2, 4, 2, 5)
print(sorted(T, reverse=True))
Output:
[5, 4, 3, 2, 2, 2, 1]
```

6: min()
It is a built-in function of the tuple. It returns the smallest element from the tuple.

```
T = (5, 8, 3, 9, 6, 4)
print(min(T))
Output:
3
```

7: max()
It is a built-in function of the tuple. It returns the largest element from the tuple.

```
T = (5, 8, 3, 9, 6, 4)
print(max(T))
Output:
9
```

8: sum()
It is a built-in function of the tuple. It returns the sum of all elements of the tuple.

```
T = (5, 8, 3, 9, 6, 4)
print(sum(T))
Output:
35
```

Nested Tuple
A nested tuple is a tuple that has been placed inside of another tuple.

```
>>> T = (5, (8, 6, 4), 3, 9)
>>> T
(5, (8, 6, 4), 3, 9)
>>> T[1]
(8, 6, 4)
>>> T[1][1]
6
```

A list can be an element of a tuple.

```
>>> T = (5, [8, 6, 4], 3, 9)
>>> T
(5, [8, 6, 4], 3, 9)

>>> T[1][2]
4
```

List is a mutable type. So we can change the value of the list, even if it is nested inside the tuple.

```
>>> T[1][2]=7
>>> T
(5, [8, 6, 7], 3, 9)
```

Suggested Programs:

Program-1 : Write a program in Python to find the maximum and minimum number from the given tuple.

```
T= (5, 8, 9, 6, 2, 5)
print("Tuple is: \n", T)
print("Minimum(Smallest) number of the Tuple = ",min(T))
print("Maximum(Largest) number of the Tuple = ",max(T))
```

Output:

```
Tuple is:
 (5, 8, 9, 6, 2, 5)
Minimum(Smallest) number of the Tuple =  2
Maximum(Largest) number of the Tuple =  9
```

Program-2 : Write a program in Python to find the mean of numeric values stored in the given tuple.

```
T=eval(input("Enter a numeric tuple: "))
print("Tuple is: \n", T)
mean=sum(T)/len(T)
print("Mean of the numeric values of the tuple = ",mean)
```

Output:

```
Enter a numeric tuple: (2, 8, 9, 6, 5)
Tuple is:
 (2, 8, 9, 6, 5)
Mean of the numeric values of the tuple =  6.0
```

Program-3 : Write a program in Python to find the index of the given number from a tuple using linear search.

```
T= (5, 8, 9, 2, 6)
print("Tuple is : ",T)
c='Y'
while c=='Y' or c=='y':
    n=int(input("Enter the number to be search in the tuple: "))
    found=False
    for i in T:
    if i==n:
        print("Element found at index: ",T.index(i))
        found=True
    if found==False:
    print("Number is not found in the list")
    c=input("Do you want to search more item (Y/N): ")
```

Output:

```
Tuple is :  (5, 8, 9, 2, 6)
Enter the number to be search in the tuple: 8
Element found at index:  1
Do you want to search more item (Y/N): y
Enter the number to be search in the tuple: 2
Element found at index:  3
Do you want to search more item (Y/N): n
```

Program-4 : Write a program in Python to count the frequency of all elements of a tuple.

```python
T= (10, 20, 30, 20, 40, 30, 20, 50)
print("Tuple is: ",T)
Uni= ()
for i in T:
     if i not in Uni:
     Uni = Uni + (i,)
for i in Uni:
     print(i,":",T.count(i))
```

Output:

```
Tuple is:  (10, 20, 30, 20, 40, 30, 20, 50)
10 : 1
20 : 3
30 : 2
40 : 1
50 : 1
```
