

Tuple

In Python programming, a tuple is similar to a list.

The difference between the two is that we cannot change the elements of a tuple once it is assigned whereas in a list, elements can be changed.

Advantages of Tuple over List

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

However, there are certain advantages of implementing a tuple over a list. Below listed are some of the main advantages:

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

Tuples is considered as sequence of immutable objects. Tuples are similar as List but it is immutable in nature.

- Heterogeneous
- Ordered
- Indexed
- Immutable
- Allows Duplicate

Consider below application which demonstrates use of Tuple

```
print("---- Marvellous Infosystems by Piyush Khairnar----")
```

```
print("Demonstration of Tuples")
```

```
tup = (11,"Marvellous",3.14,51,"Infosystems")
```

```
print(tup)
print(tup[0])
print(tup[1])
print(tup[1:])
print(tup[:2])
print(tup[1:2])
```

```
# tup[1] = "marvellous" It is not allowed to change the contents
```

```
print(len(tup))
print("Marvellous" in tup)
```

```
del tup
```

Output of above application

```
Python_Programs -- -bash -- 79x24
MacBook-Pro-de-MARVELLOUS:Python_Programs marvellous$ python Tuples.py
---- Marvellous Infosystems by Piyush Khairnar-----
Demonstration of Tuples
(11, 'Marvellous', 3.14, 51, 'Infosystems')
11
Marvellous
('Marvellous', 3.14, 51, 'Infosystems')
(11, 'Marvellous')
('Marvellous',)
5
True
MacBook-Pro-de-MARVELLOUS:Python_Programs marvellous$
```

