

## Day 14



### Tuples in Python:

#### What is a Tuple?

A tuple is a collection of multiple values stored in one variable, just like a list, but it cannot be changed.

*Tuple = immutable list*

#### Why do we need Tuples?

- To store fixed data (data that should not change)
- Faster than lists
- Safer (no accidental changes)
- Used when data should remain constant

#### Creating a Tuple

```
t1 = (1, 2, 3)
t2 = ("Python", "Java", "C")
t3 = (1, "Hello", 3.5, True)
```

#### Important: Single-Element Tuple

Not a tuple: `t = (5)`

Correct tuple: `t = (5,)`

Comma `,` makes it a tuple.

Example: a basic example of tuple.

```
1  l = (3,4,5)
2  print(l)
3  print(type(l))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL P

```
● PS E:\Python\Day11-20\Day14> python main.py
(3, 4, 5)
<class 'tuple'>
```

Example: accessing element by index.

```
1 l = (3,4,5)
2 print(l[0])
3 print(l[1])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\Python\Day11-20\Day14> python main.py
3
4
```

Example: tuples are immutable

```
5 l = (5,67,8)
6 l[0] = 45
7 print(l)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\Python\Day11-20\Day14> python main.py
Traceback (most recent call last):
  File "E:\Python\Day11-20\Day14\main.py", line 6, in <module>
    l[0] = 45
    ~~~~
TypeError: 'tuple' object does not support item assignment
```

Example: tuples can have multiple data types.

```
9 l = (3,4,5,"Aditya")
10 print(l)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS E:\Python\Day11-20\Day14> python main.py
(3, 4, 5, 'Aditya')
```

Example: negative indexing in the tuple

```
9   l = (3,4,5,"Aditya")
10  print(l[-1])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
● PS E:\Python\Day11-20\Day14> python main.py
Aditya
```

Example: to get the length of the tuple.

```
9   l = (3,4,5,"Aditya")
10  print(len(l))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
● PS E:\Python\Day11-20\Day14> python main.py
4
```

Example: looking for something in the tuple

```
12  l = (3,4,5,"Aditya",True,5.6)
13  if 4 in l:
14      print("Yes")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
● PS E:\Python\Day11-20\Day14> python main.py
Yes
```

Example: we get new tuple when we do slicing.

```
16 l = (3,4,5,6)
17 m = l[1:2]
18 print(l)
19 print(m)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\Python\Day11-20\Day14> python main.py
(3, 4, 5, 6)
(4,)
```

### Tuple vs List (Quick)

Tuple	List
Immutable	Mutable
Uses ()	Uses []
Faster	Slower
Safe	Flexible

Summary:

- Tuple stores multiple values
- Tuples are immutable
- Created using ()
- Supports indexing & slicing
- Faster and safer than lists

--The End--