

Python Data Type: TUPLE

Github: @Aaryacode19

Tuple

Tuples are used to store multiple items in a single variable.

A tuple is a collection which is ordered and **unchangeable**. Tuples are written with round brackets.

Create a Tuple:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

Tuple Items

Tuple items are ordered, unchangeable, and allow duplicate values.

Tuple items are indexed, the first item has index [0], the second item has index [1] etc.

Ordered

When we say that tuples are ordered, it means that the items have a defined order, and that order will not change.

Unchangeable

Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created.

Allow Duplicates

Since tuples are indexed, they can have items with the same value:

```
thistuple = ("apple", "banana", "cherry", "apple", "cherry")
print(thistuple)
```

Output:

```
('apple', 'banana', 'cherry', 'apple', 'cherry')
```

Tuple Length

To determine how many items a tuple has, use the len() function:

```
thistuple = ("apple", "banana", "cherry")
print(len(thistuple))
```

Output:

3

Tuple Items - Data Types

Tuple items can be of any data type:

String, int and boolean data types:

```
tuple1 = ("apple", "banana", "cherry")
tuple2 = (1, 5, 7, 9, 3)
tuple3 = (True, False, False)
```

Output:

```
('apple', 'banana', 'cherry')
(1, 5, 7, 9, 3)
(True, False, False)
```

Also a tuple can contain different data types. A tuple with strings, integers and boolean values:

```
tuple1 = ("abc", 34, True, 40, "male")
```

Access Tuple Items

You can access tuple items by referring to the index number, inside square brackets:

Range of Indexes

You can specify a range of indexes by specifying where to start and where to end the range.

When specifying a range, the return value will be a new tuple with the specified items.

Return the third, fourth, and fifth item:

```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
print(thistuple[2:5])
```

Output:

```
('cherry', 'orange', 'kiwi')
```

Join Two Tuples

To join two or more tuples you can use the `+` operator:

```
tuple1 = ("a", "b", "c")
tuple2 = (1, 2, 3)
```

```
tuple3 = tuple1 + tuple2
print(tuple3)
```

```
('a', 'b', 'c', 1, 2, 3)
```

Multiply Tuples

If you want to multiply the content of a tuple a given number of times, you can use the * operator:

Multiply the fruits tuple by 2:

```
fruits = ("apple", "banana", "cherry")  
mytuple = fruits * 2  
print(mytuple)
```

```
num = (1,2,3,4)  
mytuple1 = num * 2  
print(mytuple1)
```

Output:

```
('apple', 'banana', 'cherry', 'apple', 'banana', 'cherry')  
(1, 2, 3, 4, 1, 2, 3, 4)
```

Result: The practical has been successfully studied.