Python Programming

- Presentation By Uplatz
- Contact us: https://training.uplatz.com
- Email: info@uplatz.com
- Phone: +44 7836 212635

TUPLES

Learning outcomes:

Python Tuple
Accessing Values in Tuples
Updating Tuples
Deleting Tuple Elements
Basic Tuples Operations
Built-in Tuple Functions
Tuple Methods
Difference Between List and Tuple

Tuple

A tuple is a sequence of **immutable** Python objects. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.
A single tuple may contain **Data Types** like Integers, Strings, as well as Objects. Creating a tuple is as simple as putting different comma-separated values. Optionally you can put these comma-separated values between parentheses also.
For example:

```
tup1 = ('physics', 'chemistry', 1997, 2000);
```

Accessing Values in Tuples

To access values in tuple, use the square brackets for slicing along with the index or indices to obtain value available at that index.

For example:

```
a = (1,  12.5,  ‘a’,  “python”, “Programming”)
print(a[1])    # Output is 12.5
print(a[3])   # Output is python
print(a[1:4]) # Output is [12.5, ’a’, ‘python’]
```

Accessing Values in Tuples

Example:

```
T = (9,18,'Hi',12,"Hello",15.55,'Programming',100,125.5)
print(T[5])
print(T[1:5])
print(T[2:8])
print(T[2:9:3])
print(T[-1])
print(T[-5])
```

How to take every nth-element of a tuple?

What if we want to have only every 2-nd element of **T**? This is where the ***step parameter*** comes into play.

Uplatz

Accessing Values in Tuples

T = (9,18,'Hi',12,"Hello",15.55,'Programming',100,125.5)

print(T[0:9:3]) # Here ‘**3**’ is **step** parameter

Now you also write the above code as
print(T[::3])

Here both **print(T[0:9:3])** and **print(T[::3])** gives an output as (9, 12, 'Programming‘)

Accessing Values in Tuples

Example:

```
T = (9,18,'Hi',12,"Hello",15.55,'Programming',100,125.5)
print (T[2::2]) # ['Hi', 'Hello', 'Programming', 125.5]
print (T[2::3]) # ['Hi', 15.55, 125.5]
print (T[3::3]) #[12, 'Programming']
print(T[5:-2]) # [15.55, 'Programming']
print(T[-2:-5:-1]) # [100, 'Programming', 15.55]
print(T[-2:-7:-2]) # [100, 15.55, 12]
```

Updating Tuples

Tuples are **immutable** which means you cannot **update** or **change** the values of tuple elements. You are able to take portions of existing tuples to create new tuples as the following example demonstrates:

```
tup1 = (10, 4.6);
tup2 = (‘Hi', ‘Hello');
# Following action is not valid for tuples
# tup1[0] = 100;
# So let's create a new tuple as follows
tup3 = tup1 + tup2;
print tup3;
```

Deleting Tuple Elements

Removing individual tuple elements is not possible.
But you can delete the entire tuple.
To explicitly remove an entire tuple, just use the **del** statement.

For example:

```
tup = ('Math', 'English', 85, 65);
print (tup);
del tup;
```

Uplatz

Basic Tuples Operations

Tuples respond to the + and * operators much like strings; they mean **concatenation** and **repetition** here too, except that the result is a new tuple, not a string.

Let's see some of the basic tuple operations in Python.

Built-in Tuple Functions

Python includes the following tuple **functions**:
cmp(tuple1, tuple2) : Compares elements of both tuples.
Please note cmp() does not support in python 3.
len(tuple): Gives the total length of the tuple.
max(tuple):Returns item from the **tuple** with max value.
min(tuple): Returns item from the **tuple** with min value.
Let’s see the example of tuple functions.

Tuple Methods

Python **has two built-in methods** that you can use on tuples.
count(): Returns the number of times a specified value occurs in a tuple
index(): Searches the tuple for a specified value and returns the position of where it was found.

Let’s see the example of tuple methods.

Difference Between List and Tuple:

SR.NO.	LIST	TUPLE
1	Lists are mutable	Tuple are immutable
2	Implication of iterations is Time-consuming	Implication of iterations is comparatively Faster
3	The list is better for performing operations, such as insertion and deletion.	Tuple data type is appropriate for accessing the elements
4	Lists consume more memory	Tuple consume less memory as compared to the list
5	Lists have several built-in methods	Tuple does no have must built-in methods.

Thank you

Uplatz