Data Science with Python Programming

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

Uplatz

Tuples and Dictionary

Learning outcomes:

- Python Tuple
- Accessing, Deleting Tuple Elements
- Basic Tuples Operations
- Built-in Tuple Functions & methods
- Difference Between List and Tuple
- Python Dictionary
- Accessing, Updating, Deleting Dictionary Elements
- Built-in Functions and Methods

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:

```
t = (10,  122.5,  "A",  "Data-Science", "Python")
print(t[1])    # Output is 122.5
print(t[3])   # Output is 'Data-Science'
print(t[1:4]) # Output is (122.5, 'A', 'Data-Science')
```

Can you update Tuples element?

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;
print (tup);
```

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.

Dictionary

A dictionary is a collection of data items which are unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values.
Each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}.

Dictionary

Keys are unique within a dictionary while values may not be. **The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples**. **More than one entry per key not allowed**. Which means no duplicate key is allowed. When duplicate keys encountered during assignment, the last assignment wins .

For example:

D = {'Name': 'James', 'Age': 35, 'Degree': 'Master'};

Accessing Values in Dictionary

To access dictionary elements, you can use the familiar square brackets along with the key to obtain its value. Following is a simple example:

```
D = {'Name': 'James', 'Age': 35, 'Degree': 'Master'};
print (D['Name'])
print (D['Degree'])
print (D['Age'])
```

If we attempt to access a data item with a key, which is not part of the dictionary, we get an error

Updating Dictionary

You can update a dictionary by adding a new entry or a key-value pair, modifying an existing entry, or deleting an existing entry as shown below in the simple example:

```
D = {'Name': 'James', 'Age': 35, 'Degree': 'Master'};
D['Age'] = 28; # update existing entry
D['School'] = "DPS"; # Adding new entry
```

Delete Dictionary Elements

You can either remove individual dictionary elements or clear the entire contents of a dictionary. You can also delete entire dictionary in a single operation.

```
D = {'Name': 'James', 'Age': 35, 'Degree': 'Master'};
del D['Name'] # remove entry with key 'Name'
D.clear() # remove all entries in dict
del D # delete entire dictionary
```

Built-in Dictionary Functions and Methods

Python includes the following dictionary functions:

len(dict) : Gives the total length of the dictionary. This would be equal to the number of items in the dictionary.

str(dict) : Produces a printable string representation of a dictionary.

type(variable) : Returns the type of the passed variable. If passed variable is dictionary, then it would return a dictionary type.

Built-in Dictionary Functions and Methods

Python includes following dictionary methods:

dict.clear() : Removes all elements of dictionary *dict.*

dict.copy() : Returns a shallow copy of dictionary *dict*

dict.fromkeys() : Create a new dictionary with keys from seq and values *set* to *value*.

dict.get(key, default=None) : For key key, returns value or default if key not in dictionary

Built-in Dictionary Functions and Methods

Python includes following dictionary methods:

dict.items() : Returns a list of *dict*'s (key, value) tuple pairs.

dict.keys() : Returns list of dictionary dict's keys

dict.setdefault(key, default=None) :
Similar to get(), but will set dict[key]=default if *key* is not already in dict

dict.update(dict2) : Adds dictionary *dict2*'s key-values pairs to *dict*

dict.values() : Returns list of dictionary *dict*'s values