

Python Programming

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

DICTIONARY

Learning outcomes:

Python Dictionary

Accessing Values in Dictionary

Updating Dictionary

Delete Dictionary Elements

Properties of Dictionary Keys

**Built-in Dictionary Functions and
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.

For example:

```
D = {'Name': 'John', 'Age': 25, 'Degree': 'Bachelor'};
```

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': 'John', 'Age': 25, 'Degree': 'Bachelor'}  
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': 'John', 'Age': 25, 'Degree':  
'Bachelor', 'Marks': 75}
```

```
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': 'John', 'Age': 25, 'Degree':  
'Bachelor', 'Marks': 75}
```

```
del D['Name'] # remove entry with key 'Name'
```

```
D.clear() # remove all entries in dict
```

```
del D # delete entire dictionary
```


Properties of Dictionary Keys

Dictionary values have no restrictions. They can be any arbitrary Python object, either standard objects or user-defined objects. However, same is not true for the keys.

There are two important points to remember about dictionary keys:

1) 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

Properties of Dictionary Keys

2) Keys must be immutable. Which means you can use strings, numbers or tuples as dictionary keys but something like ['key'] is not allowed.

Let's see the example.

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

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



Thank you