Lecture#7 Python Programming

Python Dictionary Append: How to Add Key/Value Pair

Dictionary is one of the important data types available in Python. The data in a dictionary is stored as a key/value pair. It is separated by a colon(:), and the key/value pair is separated by comma(,).
The keys in a dictionary are unique and can be a string, integer, tuple, etc. The values can be a list or list within a list, numbers, string, etc.

Here is an example of a dictionary:

```
my_dict = {"a": A, "b": B, "c": C, "d": D}
```

In this Python tutorial, you will learn:

- **Restrictions on Key Dictionaries**
- **How to append an element to a key in a dictionary with Python?**
- **Accessing elements of a dictionary**
- **Deleting element(s) in a dictionary**
- **Deleting Element(s) from dictionary using pop() method**
- **Appending element(s) to a dictionary**
- **Updating existing element(s) in a dictionary**
- **Insert a dictionary into another dictionary**

Restrictions on Key Dictionaries

Here is a list of restrictions on the key in a dictionary:

- If there is a duplicate key defined in a dictionary, the last is considered. For example consider dictionary my_dict = {“Name”:”ABC”,”Address”:”Mumbai”,”Age”:30, “Name”: “XYZ”};.It has a key

"Name" defined twice with value as ABC and XYZ. The preference will be given to the last one defined, i.e., "Name": "XYZ."

- The data-type for your key can be a number, string, float, boolean, tuples, built-in objects like class and functions.For example my_dict = {bin:"001", hex:"6" ,10:"ten", bool:"1", float:"12.8", int:1, False:'0'};Only thing that is not allowed is, you cannot defined a key in square brackets for example my_dict = {["Name"]:"ABC","Address":"Mumbai","Age":30};

How to append an element to a key in a dictionary with Python?

We can make use of the built-in function append() to add elements to the keys in the dictionary. To add element using append() to the dictionary, we have first to find the key to which we need to append to.

Consider you have a dictionary as follows:

```
my_dict = {"Name":[],"Address":[],"Age":[]};
```

The keys in the dictionary are Name, Address and Age. Usingappend() methodwe canupdate the values for the keys in the dictionary.

```
my_dict = {"Name":[],"Address":[],"Age":[]};

my_dict["Name"].append("Guru")
my_dict["Address"].append("Mumbai")
my_dict["Age"].append(30)
print(my_dict)
```

When we print the dictionary after updating the values, the output is as follows:

Output:

```
{'Name': ['Guru'], 'Address': ['Mumbai'], 'Age': [30]}
```

Accessing elements of a dictionary

The data inside a dictionary is available in a key/value pair. To access the elements from a dictionary, you need to use square brackets ([‘key’]) with the key inside it.

Here is an example that shows to accesselements from the dictionary by using the key in the square bracket.

```
my_dict = {"username": "XYZ", "email": "xyz@gmail.com", "location":"Mumbai"}
print("username :", my_dict['username'])
print("email : ", my_dict["email"])
print("location : ", my_dict["location"])
```

Output:

```
username : XYZ
email :  xyz@gmail.com
location :  Mumbai
```

If you try to use a key that is not existing in the dictionary , it will throw an error as shown below:

```
my_dict = {"username": "XYZ", "email": "xyz@gmail.com", "location":"Mumbai"}
print("name :", my_dict['name'])
```

Output:

```
Traceback (most recent call last):
File "display.py", line 2, in <module>
print("name :", my_dict['name'])
KeyError: 'name'
```

Deleting element(s) in a dictionary

To delete an element from a dictionary, you have to make use of the **del** keyword.

The syntax is :

```
del dict['yourkey']  # This will remove the element with your key.
```

To delete the entire dictionary, you again can make use of the del keyword as shown below:

```
del my_dict  # this will delete the dictionary with name my_dict
```

To just empty the dictionary or clear the contents inside the dictionary you can makeuse of clear() method on your dictionaryas shown below:

```
your_dict.clear()
```

Here is a working example that shows the deletion of element, to clear the dict contents and to delete entire dictionary.

```
my_dict = {"username": "XYZ", "email": "xyz@gmail.com",
"location":"Mumbai"}
del my_dict['username']  # it will remove "username": "XYZ" from my_dict
print(my_dict)
my_dict.clear()  # till will make the dictionarymy_dictempty
print(my_dict)
delmy_dict # this will delete the dictionarymy_dict
print(my_dict)
```

Output:

```
{'email': 'xyz@gmail.com', 'location': 'Mumbai'}
{}
Traceback (most recent call last):
  File "main.py", line 7, in <module>
    print(my_dict)
NameError: name 'my_dict' is not defined
```

Deleting Element(s) from dictionary using pop() method

In addition to the del keyword, you can also make use of dict.pop() method to remove an element from the dictionary. The pop() is a built-in method available with a dictionary that helps to delete the element based on the key given.

Syntax:

```
dict.pop(key, defaultvalue)
```

The pop() method returns the element removed for the given key, and if the given key is not present, it will return the defaultvalue. If the defaultvalue is not given and the key is not present in the dictionary, it will throw an error.

Here is a working example that shows using of dict.pop() to delete an element.

```
my_dict = {"username": "XYZ", "email": "xyz@gmail.com", "location":"Mumbai"}
my_dict.pop("username")
print(my_dict)
```

Output:

```
{'email': 'xyz@gmail.com', 'location': 'Mumbai'}
```

Appending element(s) to a dictionary

To append an element to an existing dictionary, you have to use the dictionary name followed by square brackets with the key name and assign a value to it.

Here is an example of the same:

```
my_dict = {"username": "XYZ", "email": "xyz@gmail.com", "location":"Mumbai"}

my_dict['name']='Nick'

print(my_dict)
```

Output:

```
{'username': 'XYZ', 'email': 'xyz@gmail.com', 'location': 'Mumbai', 'name':
'Nick'}
```

Updating existing element(s) in a dictionary

To update the existing elements inside a dictionary, you need a reference to the key you want the value to be updated.

So we have a dictionary my_dict = {“username”: “XYZ”, “email”: “xyz@gmail.com”, “location”:”Mumbai”}.

We would like to update the **username** from XYZ to ABC . Here is an example that shows how you can update it.

```
my_dict = {"username": "XYZ", "email": "xyz@gmail.com",
"location":"Mumbai"}

my_dict["username"] = "ABC"

print(my_dict)
```

Output:

```
{'username': 'ABC', 'email': 'xyz@gmail.com', 'location': 'Mumbai'}
```

Insert a dictionary into another dictionary

Consider you have two dictionaries as shown below:

Dictionary 1:

```
my_dict = {"username": "XYZ", "email": "xyz@gmail.com",
"location":"Washington"}
```

Dictionary 2:

```
my_dict1 = {"firstName" : "Nick", "lastName": "Price"}
```

Now I want my_dict1 dictionary to be inserted into my_dict dictionary. To do that lets create a key called “name” in my_dict and assign my_dict1 dictionary to it.

Here is a working example that shows inserting my_dict1 dictionary into my_dict.

```
my_dict = {"username": "XYZ", "email": "xyz@gmail.com",
"location":"Washington"}

my_dict1 = {"firstName" : "Nick", "lastName": "Price"}

my_dict["name"] = my_dict1

print(my_dict)
```

Output:

```
{'username': 'XYZ', 'email': 'xyz@gmail.com', 'location': 'Mumbai', 'name':
{'firstName': 'Nick', 'lastName': 'Price'}}
```

Now if you see the key “name”, it has the dictionary my_dict1.

Summary:

- Dictionary is one of the important data types available in Python. The data in a dictionary is stored as a key/value pair. The key/value is separated by a colon(:), and the key/value pair is separated by comma(,). The keys in a dictionary are unique and can be a string, integer, tuple, etc. The values can be a list or list within a list, numbers, string, etc.

Important built-in methods on a dictionary:

Method	Description
clear()	It will remove all the elements from the dictionary.
append()	It is a built-in function in Python that helps to update the values for the keys in the dictionary.
update()	The update() method will help us to merge one dictionary with another.
pop()	Removes the element from the dictionary.