

Course Name: Python Programming with Django

Name: Bipul Dutta

Batch ID: 13328

Order No: 4099423

Assignment- 05.

Date: 05.04.22

**** Make a note on Dictionary data type.**

Python Dictionary Data Type:

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in perl and consist of key- value pairs. A dictionary key can be almost any python type, but are usually numbers or strings. Values , on the other hand, can be any arbitrary python object.

Creating Python Dictionary:

Dictionaries are enclosed by curly braces {} and values can be assigned and accessed using square braces [], and each key is separated from its value by the colon (:). For example:

```
dict = {}
print(type(dict))
dict = {
    'Name' : 'Bipul Dutta',
    'Address': 'Dhaka',
}
print(dict)
print(dict)

#output
#<class 'dict'>
#{'Name': 'Bipul Dutta', 'Address': 'Dhaka'}
```

Coding for update item:

```
dict = {
    'Name' : 'Bipul Dutta',
    'Address': 'Dhaka',
}
print(dict)
dict['Name'] = 'Dutta Bipul'
print(dict)

#output
#{'Name': 'Bipul Dutta', 'Address': 'Dhaka'}
#{'Name': 'Dutta Bipul', 'Address': 'Dhaka'}
```

Adding elements in Dictionary:

```
dict = {
    'Name': 'Bipul Dutta',
    'Address': 'Dhaka',
}
print(dict)
dict2 = {
    'Home Town': 'Khulna',
    'Mobile' : '01846291082'
}
dict.update(dict2)
print(dict)

#output
#{'Name': 'Bipul Dutta', 'Address': 'Dhaka'}
#{'Name': 'Bipul Dutta', 'Address': 'Dhaka', 'Home Town': 'Khulna', 'Mobile': '01846291082'}
```

Delete Key from Dictionary:

```
dict = {
    'Name': 'Bipul Dutta',
    'Address': 'Dhaka',
}
print(dict)
dict2 = {
    'Home Town': 'Khulna',
    'Mobile' : '01846291082'
}
dict.update(dict2)
print(dict)
del dict['Address']
print(dict)

#output
#{'Name': 'Bipul Dutta', 'Address': 'Dhaka', 'Home Town': 'Khulna', 'Mobile': '01846291082'}
#{'Name': 'Bipul Dutta', 'Home Town': 'Khulna', 'Mobile': '01846291082'}
```

Python Dictionary Methods:

Methods that are available with a dictionary are tabulated below. Some of them have already been used in the above examples:

Method	Description
Clear()	Removes all items from the dictionary
Copy()	Returns a shallow copy of the dictionary
Get()	Returns the value of the key. If the key does not exist, returns.
Pop()	Removes the item with the key and returns its value if key is not found. If d is not provided and the key is not found, it raises <code>keyError</code> .
Update()	Updates the dictionary with the key/ value pairs from other, overwriting existing keys.
Keys()	Returns a new objects of the dictionary's keys.

More others.

Dictionary Built-in Functions:

Built-in functions like `all()`, `any()`, `len()`, `cmp()`, `sorted()` etc are commonly used with dictionaries to perform different tasks.

Function	Description
All()	Return <code>True</code> if all keys of the dictionary are <code>True</code> (or if the dictionary is empty).
Any()	Return <code>True</code> if any key of the dictionary is <code>true</code> . If the dictionary is empty, return <code>False</code> .
Len()	Return the length (the number of items) in the dictionary.
Cmp()	Compares items of two dictionaries. (Not available in Python 3)
Sorted()	Return a new sorted list of keys in the dictionary.