Dictionary

- ☐ A dictionary is an unordered collection of items.
- ☐ Each item of a dictionary has a key and corresponding value associated to it (key: value).
- ☐ keys must be immutable (i.e. can be string, number or tuple) and must be unique.
- ☐ A dictionary is created by placing all the items (key:value) inside curly braces {} separated by commas.

```
>>>emptyDict = {}
>>>print("Empty Dictionary:",emptyDict)
Empty Dictionary: {}
>>>Dict = {"Name": "XYZ", "Roll No.": "B19001" ,"Branch":
    "Data Science","Age": 19}
>>>print(Dict)
{'Name': 'XYZ', 'Roll No.': 'B19001', 'Branch': 'Data
    Science', 'Age': 19}
```

Adding and Updating a Key-Value Pair

```
>>>Dict = {}
#adding key-value pair
>>>Dict["IC110"] = 4
>>>Dict["IC152"] = 4
>>>Dict["IC152P"] = 1
>>>Dict["Nested_Dict"] = {"IC140": 4, "IC160" : 4}
>>>print(Dict)
{'IC110': 4, 'IC152': 4, 'IC152P': 1, 'Nested_Dict':
    {'IC140': 4, 'IC160': 4}}
#Accesing values from nested dictionary
>>>print(Dict["Nested_Dict"]["IC160"])
4
#updating value
>>>Dict["IC152"] = 3
>>>print(Dict)
{'IC110': 4, 'IC152': 3, 'IC152P': 1, 'Nested_Dict':
    {'IC140': 4, 'IC160': 4}}
```

Accessing Values from a Dictionary

- ☐ The values of dictionary can be accessed by using key name inside square brackets or inside get() function.

```
>>>Dict = {"Name": "XYZ", "Roll No.": 'B19001',"Branch":
    "Data Science","Age": 19}
>>>print(Dict["Name"])
XYZ
>>>print(Dict.get("Name"))
XYZ
>>>print(Dict.get("Year"))
None
>>>print(Dict["Year"])
KeyError: 'Year'
```

Dictionary Methods

Method	Explanation
keys()	**Returns view of all the keys in dictionary**
values()	**Returns a view of all the values in dictionary**
update()	**Adds dictionary key-values pairs (passed as argument) to the dictionary which call update() method**
pop()	**Removes and returns an element from a dictionary having the given key.**

Dictionary Methods

popitem()	**Removes the arbitrary key-value pair from the dictionary and returns it as tuple.**
copy()	**Returns a shallow copy of the dictionary**
clear()	**Removes all the items from dictionary.**

values()

- values() methods returns a view object that displays all the values in dictionary.

Syntax

dict.values()

```
>>>Courses_Dict = {"IC110": 4, "IC152": 4, "IC152P": 1,
    "Nested_Dict": {"IC140": 4, "IC160" : 4}}
>>>Values = Courses_Dict.values()
>>>print(Values)
dict_values([4, 4, 1, {'IC140': 4,'IC160': 4}])
>>>del Courses_Dict["Nested_Dict"]
>>>print(Values)
dict_values([4, 4, 1])
>>>Values = list(Courses_Dict.Values())
>>>del Courses_Dict["IC152"]
>>>print(Values)
[4, 4, 1]
```

keys()

- keys() methods returns a view object that displays all the keys.
- This view object changes according to the changes in the dictionary.
- view object can be changed to list or tuple by using typecasting.

```
>>>Courses_Dict = {"IC110": 4, "IC152": 4, "IC152P": 1,
    "Nested_Dict": {"IC140": 4, "IC160" : 4}}
>>>Keys = Courses_Dict.keys()
>>>print(Keys)
dict_keys(['IC110', 'IC152', 'IC152P','Nested_Dict'])
>>>del Courses_Dict["IC152"]
>>>print(Keys)
dict_keys(['IC110', 'IC152P','Nested_Dict'])
>>>Keys = list(Courses_Dict.keys())
>>>del Courses_Dict["IC152P"]
>>>print(Keys)
['IC110', 'IC152P','Nested_Dict']
```

update()

- update() method updates the dictionary with the elements from the another dictionary.

Syntax

dict1.update(dict2)

```
>>>Courses_Dict1 = {"IC110": 4, "IC152": 4, "IC152P": 1}
>>>Courses_Dict2 = {"IC140": 4, "IC152": 3, "IC160": 4}
>>>Courses_Dict1.update(Courses_Dict2)
>>>print(Courses_Dict1)
{'IC110': 4,'IC152': 3, 'IC152P': 1, 'IC140': 4, 'IC160':
   4}
```

pop()

- ☐ pop() method removes and returns an element from a dictionary having the given key.

Syntax

dict.pop(key_name)

```
>>>Courses_Dict = {"IC110": 4, "IC152": 4, "IC152P": 1,
    "Nested_Dict": {"IC140": 4, "IC160" : 4}}
>>>element = Courses_Dict.pop("Nested_Dict")
>>>print(element)
{'IC140': 4, 'IC160': 4}
>>>print(Courses_Dict)
{'IC110': 4,'IC152': 4, 'IC152P': 1}
>>>element = Courses_Dict.pop("IC140")
KeyError: 'IC140'
```

copy()

- ☐ copy() method returns a shallow copy of the dictionary.

Syntax

dict.copy()

```
>>>Courses_Dict = {"IC110": 4, "IC152": 4, "IC152P": 1}
>>>New_Dict = Courses_Dict.copy()
>>>print(New_Dict)
{'IC110': 4, 'IC152': 4, 'IC152P': 1}
>>>print(id(Courses_Dict) == id(New_Dict))
False
```

popitem()

- ☐ popitem() returns and removes an arbitrary key-value pair from the dictionary.

Syntax

dict.popitem()

```
>>>Courses_Dict = {"IC110": 4, "IC152": 4, "Nested_Dict":
    {"IC140": 4, "IC160" : 4},  "IC152P": 1}
>>>pair = Courses_Dict.popitem()
>>>print(pair)
('IC152P', 1)
>>>print(Courses_Dict)
{'IC110': 4, 'IC152': 4, 'Nested_Dict': {'IC140': 4,
    'IC160': 4}}
```

clear()

- ☐ clear() method removes all the items form dictionary.

Syntax

dict.clear()

```
>>>Courses_Dict = {"IC110": 4, "IC152": 4, "IC152P": 1}
>>>Courses_Dict.clear()
>>>print(Courses_Dict)
{}
```