Assignment Set

Dictionary

NAME : Lovenish Gaur

Dictionary

A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values.

1. clear() - Removes all the elements from the dictionary
2. copy() -Returns a copy of the dictionary
3. fromkeys() - Returns a dictionary with the specified keys and value
4. get()- Returns the value of the specified key
5. items()- Returns a list containing a tuple for each key value pair
6. keys() - Returns a list containing the dictionary's keys
7. pop()- Removes the element with the specified key
8. popitem()- Removes the last inserted key-value pair
9. setdefault()- Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
10. update()- Updates the dictionary with the specified key-value pairs
11. values() - Returns a list of all the values in the dictionary

In [11]:

```
dict = {
  "Company": "USA",
  "Speciality": "Mustang",
  "Location": "Netherland",
    "Founder": "Henry Ford",
    "Year" : 1989,
    "Revenue" : 417323500000.00
}
dict
```

Out[11]:

```
{'Company': 'USA',
 'Speciality': 'Mustang',
 'Location': 'Netherland',
 'Founder': 'Henry Ford',
 'Year': 1989,
 'Revenue': 417323500000.0}
```

In [12]:

```
#copy() -Returns a copy of the dictionary
Details = dict.copy()
Details
```

Out[12]:

```
{'Company': 'USA',
 'Speciality': 'Mustang',
 'Location': 'Netherland',
 'Founder': 'Henry Ford',
 'Year': 1989,
 'Revenue': 417323500000.0}
```

In [30]:

```
#fromkeys() - Returns a dictionary with the specified keys and value
Name = ("Maruti","Mahindra","Ford","Ferrari","Corvette","Hundayi","Volkswagon")
Average =(23.4,20.8)
Cars_Average = dict.fromkeys(Name,Average)
Cars_Average
```

Out[30]:

```
{'Maruti': (23.4, 20.8),
 'Mahindra': (23.4, 20.8),
 'Ford': (23.4, 20.8),
 'Ferrari': (23.4, 20.8),
 'Corvette': (23.4, 20.8),
 'Hundayi': (23.4, 20.8),
 'Volkswagon': (23.4, 20.8)}
```

In [33]:

```
#get()- Returns the value of the specified key
Test1 = Cars_Average.get("Maruti")
Test1
```

Out[33]:

```
(23.4, 20.8)
```

In [35]:

```
# items()- Returns a list containing a tuple for each key value pair
new = Cars_Average.items()
new
```

Out[35]:

```
dict_items([('Maruti', (23.4, 20.8)), ('Mahindra', (23.4, 20.8)), ('Fo
rd', (23.4, 20.8)), ('Ferrari', (23.4, 20.8)), ('Corvette', (23.4, 20.
8)), ('Hundayi', (23.4, 20.8)), ('Volkswagon', (23.4, 20.8))])
```

In [36]:

```
#keys() - Returns a list containing the dictionary's keys
new1 = Cars_Average.keys()
new1
```

Out[36]:

```
dict_keys(['Maruti', 'Mahindra', 'Ford', 'Ferrari', 'Corvette', 'Hunda
yi', 'Volkswagon'])
```

In [37]:

```
#pop()- Removes the element with the specified key
new3 = Cars_Average.pop("Hundayi")
new3
```

Out[37]:

```
(23.4, 20.8)
```

In [38]:

```
Cars_Average
```

Out[38]:

```
{'Maruti': (23.4, 20.8),
 'Mahindra': (23.4, 20.8),
 'Ford': (23.4, 20.8),
 'Ferrari': (23.4, 20.8),
 'Corvette': (23.4, 20.8),
 'Volkswagon': (23.4, 20.8)}
```

In [39]:

```
#popitem()- Removes the last inserted key-value pair
new4 = Cars_Average.popitem()
new4
```

Out[39]:

```
('Volkswagon', (23.4, 20.8))
```

In [40]:

```
Cars_Average
```

Out[40]:

```
{'Maruti': (23.4, 20.8),
 'Mahindra': (23.4, 20.8),
 'Ford': (23.4, 20.8),
 'Ferrari': (23.4, 20.8),
 'Corvette': (23.4, 20.8)}
```

In [54]:

```
#setdefault()- Returns the value of the specified key.
#If the key does not exist: insert the key, with the specified value

new5 = Cars_Average.setdefault("Maruti" ,(25.6,23.4))
new5
```

Out[54]:

```
(23.4, 20.8)
```

In [63]:

```
#setdefault()-update()- Updates the dictionary with the specified key-value pairs
new6 = Cars_Average.update({"Ford":(25.7,23.4)})
Cars_Average
```

Out[63]:

```
{'Maruti': (23.4, 20.8),
 'Mahindra': (23.4, 20.8),
 'Ford': (25.7, 23.4),
 'Ferrari': (23.4, 20.8),
 'Corvette': (23.4, 20.8)}
```

In [65]:

```
#values() - Returns a list of all the values in the dictionary
new7 = Cars_Average.values()
new7
```

Out[65]:

```
dict_values([(23.4, 20.8), (23.4, 20.8), (25.7, 23.4), (23.4, 20.8),
(23.4, 20.8)])
```

In [ ]: