Dictionaries

- Dictionary is an unordered collection.

empty dictionary

In [1]:

```
empty_dict_1 = {}
print(empty_dict_1)

empty_dict_2 = dict()
print(empty_dict_2)
```

```
{}
{}
```

- In Python, dictionaries are defined within braces {} with each item being **key:value**
- **keys can be of fundamental data type** - Generally, the keys are int or str
- **values can be of any data type**

dictionary with 1 item

In [2]:

```
d1={1:4}
print(d1)
```

```
{1: 4}
```

dictionary with 2 items

In [3]:

```
d2 = {1:455,"a":45.6}
d2
```

Out[3]:

```
{1: 455, 'a': 45.6}
```

dictionary with multiple items

In [4]:

```
d = {1: 456, 10.4: [3,9,15], 'key3': (94,8,23), 'key4': {'item1','item2','item3'}}
d
```

Out[4]:

```
{1: 456,
 10.4: [3, 9, 15],
 'key3': (94, 8, 23),
 'key4': {'item1', 'item2', 'item3'}}
```

convert a list of tuples into a dictionary

In [5]:

```
family = dict([('dad', 'business'), ('mom', 'house_wife'), ('size', 6)])
family
```

Out[5]:

```
{'dad': 'business', 'mom': 'house_wife', 'size': 6}
```

Nested Dictionary

- If atleast one value is dict, then it is called as nested dictionary

In [6]:

```
cars={'car1':1989,'car2':[1,2,3],'car3':{110:3}}
print(cars)
```

```
{'car1': 1989, 'car2': [1, 2, 3], 'car3': {110: 3}}
```

indexing in dictionaries

- **Syntax:** dict[key name]

In [7]:

```
cars["car1"]
```

Out[7]:

```
1989
```

In [8]:

```
cars["car2"][0]
```

Out[8]:

```
1
```

In [9]:

```
cars["car3"][110]
```

Out[9]:

```
3
```

Dictionary Methods

In [10]:

```
marks={'history':45,'Hindi':56}
marks
```

Out[10]:

```
{'history': 45, 'Hindi': 56}
```

dict.items()

- **items()** method returns a list of key-value pairs in the dictionary

In [11]:

```
marks.items()
```

Out[11]:

```
dict_items([('history', 45), ('Hindi', 56)])
```

dict.keys()

- **keys()** method returns a list of keys in the dictionary

In [12]:

```
marks.keys()
```

Out[12]:

```
dict_keys(['history', 'Hindi'])
```

dict.values()

- **values()** method returns a list of values in the dictionary

In [13]:

```
marks.values()
```

Out[13]:

```
dict_values([45, 56])
```

to add a single item into a dictionary

- option 1

In [14]:

```
marks={'history':45,'Hindi':56}

marks['english']=47
print(marks)
```

```
{'history': 45, 'Hindi': 56, 'english': 47}
```

- option 2

In [15]:

```
marks={'history':45,'Hindi':56}

marks.update({'english':47})
print(marks)
```

```
{'history': 45, 'Hindi': 56, 'english': 47}
```

to add single or mutiple items into a dictionary

In [16]:

```
marks.update({'Chemistry':89,'Physics':98})
print(marks)
```

```
{'history': 45, 'Hindi': 56, 'english': 47, 'Chemistry': 89, 'Physics': 9
8}
```

to delete item in dictionary

- option 1 : del dict[key]

In [17]:

```
marks={'history':45, 'Hindi':56}

del marks['history']
print(marks)
```

```
{'Hindi': 56}
```

- option 2 : **dict.pop(key)**
- **pop()** method removes and return the value of the specified key.
- If the key does not exist, it raises a KeyError

In [18]:

```
marks={'history':45,'Hindi':56}

marks.pop("history")
print(marks)
```

```
{'Hindi': 56}
```

dict.clear()

- **clear()** method removes all items from the dictionary

In [19]:

```
marks={'history':45,'Hindi':56}
marks.clear()

print(marks)
```

```
{}
```

dict.copy()

- copy() method returns a shallow copy of the dictionary
- changes to the copy of the dictionarydo not affect the original dictionary

In [20]:

```
d ={'history':45,'Hindi':56}

a = d.copy()
a.update({"maths":99})
print(a)

print(d)
```

```
{'history': 45, 'Hindi': 56, 'maths': 99}
{'history': 45, 'Hindi': 56}
```

dict.get()

- **get()** method returns the value of the specified key.
- If the key is not present in the dictionary, it will return "None"
- we can get the value by using another way also **dict_name[key]**

In [21]:

```
marks={'history':45,'Hindi':56}

marks.get("history")
```

Out[21]:

```
45
```

In [22]:

```
marks={'history':45,'Hindi':56}

marks['history']
```

Out[22]:

```
45
```

to replace value in dictionary

In [23]:

```
marks={'history':45, 'Hindi':56}

marks['hindi']=64
marks
```

Out[23]:

```
{'history': 45, 'Hindi': 56, 'hindi': 64}
```

to replace key in dictionary

In [24]:

```
marks={'history':45, 'Hindi':56}

marks['Maths'] = marks.pop('Hindi')
print(marks)
```

```
{'history': 45, 'Maths': 56}
```