Loops in Dictionary

- Looping through a dictionary is possible in python it gives output in the form of keys

```
>>>
>>> dict1 ={ 101 : 'Production' , 102 : 'Accounts' , 103 : 'Sales & Marketing' , 104 : 'Inventory' }
>>>
>>> for x in dict1:
        print(x)


101
102
103
104
```

- If you want the key-value as output then we can loop like this

```
>>> for x in dict1:
        print(x, dict1[x])


101 Production
102 Accounts
103 Sales & Marketing
104 Inventory
>>>
```

- To an index when you give a key it gives output in key - value
- We can also use get(x) method to get the result

```
>>> for x in dict1:
        print(x, dict1.get(x) )


101 Production
102 Accounts
103 Sales & Marketing
104 Inventory
>>>
```

- The difference between dict[x] and get[x] is when there is no key present in dictionary but you still call it then dict[x] will given an error while get[x] will not give an error

```
>>>
>>> print(dict1[102])
Accounts
>>> print(dict1.get(102))
Accounts
>>> print(dict1[106])
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in <module>
    print(dict1[106])
KeyError: 106
>>> print(dict1.get(106))
None
>>> print(dict1.get(106,'Unkown Dept')
    )
Unkown Dept
>>>
```

- Some other methods on dictionary which can be use in looping are keys( ) , value( ) and item( )

```
>>> dict1 = { 101 : 'Production' , 102 :'Accounts' , 103 : 'Sales & Marketing' , 104 : 'Inventory' }
>>>
>>> dict1.keys()
dict_keys([101, 102, 103, 104])
>>> dict1.values()
dict_values(['Production', 'Accounts', 'Sales & Marketing', 'Inventory'])
>>> dict1.items()
dict_items([(101, 'Production'), (102, 'Accounts'), (103, 'Sales & Marketing'), (104, 'Inventory')])
>>>
>>> for k in dict1.keys():
        print(k, dict1[k])


101 Production
102 Accounts
103 Sales & Marketing
104 Inventory
>>>
>>> for v in dict1.values():
        print(v)
```