**How to read content of dictionary?**
Dictionary content can be read in different ways

1. Using key
2. Using for loop
3. Using dict functions
    a. Keys()
    b. Values()
    c. Items()
    d. get()

**Reading content of dictionary using key**
Dictionary is key based collection, the content of dictionary is read using key.

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

**If key exists, return its value**
**If key not exists generate KeyError**

**Example**

```
>>> courses={'java':2000,
...          'python':4000,
...          'oracle':1000}
>>> print(courses)
{'java': 2000, 'python': 4000, 'oracle': 1000}
>>> courses['python']
4000
>>> courses['java']
2000
>>> courses['c++']
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    courses['c++']
KeyError: 'c++'
```

**Example:**

```
# Login Application
```

```
users={'nit':'nit123',
       'ramesh':'ram123',
       'suresh':'s321'}

uname=input("UserName ") # nit
pwd=input("Password ") # xyz

if uname in users:
    p=users[uname]
    if p==pwd:
        print(f'{uname} Welcome')
    else:
        print("invalid password")
else:
    print("invalid username")
```

**Output:**

```
UserName suresh
Password s321
suresh Welcome

UserName nit
Password xyz
invalid password

UserName naresh
Password nit123
invalid username
```

**Using for loop**

For loop is an iterator, which iterate or read keys from dictionary.

**Example:**

```
d1={1:10,2:20,3:30,4:40,5:50}
for a in d1:
    print(a,d1[a])
```

**Output:**

```
1 10
2 20
3 30
4 40
5 50
```

**Example:**

```
sales={2000:45000,
       2001:54000,
       2002:90000,
       2003:50000,
       2004:95000}


tot=0
for year in sales:
    s=sales[year]
    tot=tot+s

print(f'Sales {sales}')
print(f'Total Sales {tot}')
```

**Output:**

```
Sales {2000: 45000, 2001: 54000, 2002: 90000, 2003: 50000, 2004:
95000}
Total Sales 334000
```

**Example:**

```
marks={101:[50,60,70],
       102:[60,70,80],
       103:[40,30,60]}
```

```
print(marks[101])
print(marks[102])
print(marks[103])
print(marks[101][0],marks[101][1],marks[101][2])
```

**Output:**

```
[50, 60, 70]
[60, 70, 80]
[40, 30, 60]
50 60 70
```

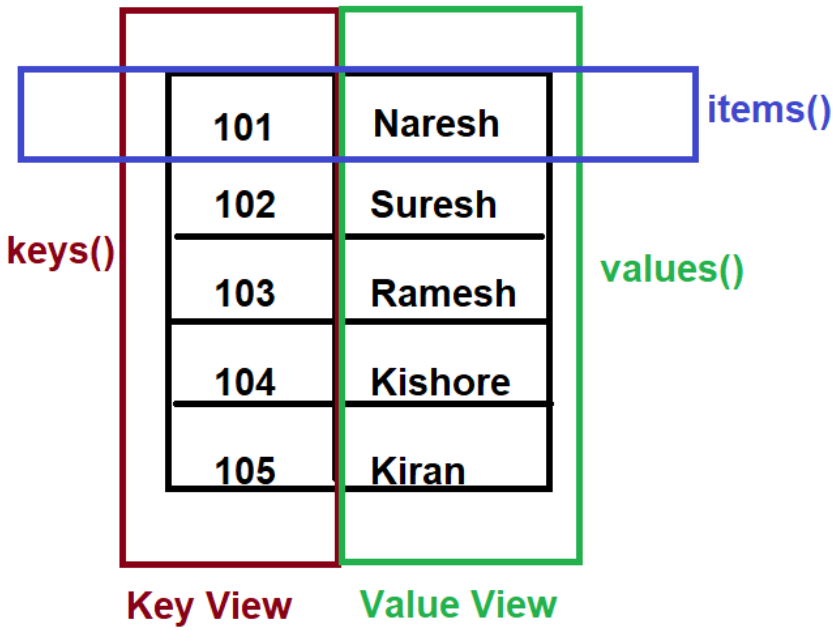
**Example:**

```
marks={101:{'sub1':40,'sub2':60,'sub3':70},
       102:{'sub1':90,'sub2':80,'sub3':50},
       103:{'sub1':60,'sub2':70,'sub3':90}}

print(marks[101])
print(marks[102])
print(marks[103])
print(marks[101]['sub1'],marks[101]['sub2'],marks[101]['sub3'])
print(marks[102]['sub1'],marks[102]['sub2'],marks[102]['sub3'])
print(marks[103]['sub1'],marks[103]['sub2'],marks[103]['sub3'])

for rno in marks:
    mdata=marks[rno]
    for k in mdata:
        print(mdata[k],end=' ')
    print()
```

**Output:**

```
{'sub1': 40, 'sub2': 60, 'sub3': 70}
{'sub1': 90, 'sub2': 80, 'sub3': 50}
{'sub1': 60, 'sub2': 70, 'sub3': 90}
40 60 70
90 80 50
60 70 90
40 60 70
90 80 50
60 70 90
```

**Dictionary view objects**
The objects returned by dict.keys(), dict.values() and dict.items() are *view objects*. They provide a dynamic view on the dictionary's entries, which means that when the dictionary changes, the view reflects these changes.




**Example:**

```
email_dict={'naresh':'naresh@nareshit.com',
            'suresh':'suresh@gmail.com',
            'kishore':'kishore@gmail.com',
            'kiran':'kiran@nareshit.com'}

# Creating key View
names=email_dict.keys()
print(names)
for name in names:
    print(name)
# Creating values view
email_ids=email_dict.values()
print(email_ids)
for emailid in email_ids:
    print(emailid)
# Creating items view
```

```
persons=email_dict.items()
print(persons)
for person in persons:
    print(person)
    name,email=person
    print(name,email)
```

**Output:**

```
dict_keys(['naresh', 'suresh', 'kishore', 'kiran'])
naresh
suresh
kishore
kiran
dict_values(['naresh@nareshit.com', 'suresh@gmail.com',
'kishore@gmail.com', 'kiran@nareshit.com'])
naresh@nareshit.com
suresh@gmail.com
kishore@gmail.com
kiran@nareshit.com
dict_items([('naresh', 'naresh@nareshit.com'), ('suresh',
'suresh@gmail.com'), ('kishore', 'kishore@gmail.com'), ('kiran',
'kiran@nareshit.com')])
('naresh', 'naresh@nareshit.com')
naresh naresh@nareshit.com
('suresh', 'suresh@gmail.com')
suresh suresh@gmail.com
('kishore', 'kishore@gmail.com')
kishore kishore@gmail.com
('kiran', 'kiran@nareshit.com')
kiran kiran@nareshit.com
```

**get(key[, default])**

Return the value for key if key is in the dictionary, else default. If default is not given, it defaults to None, so that this method never raises a KeyError.

**Example:**

```
d1={1:10,2:20,3:30,4:40,5:50}

value1=d1.get(1)
value2=d1.get(4)
```

```
value3=d1.get(6)
print(value1,value2,value3)
value4=d1.get(6,0)
print(value4)
```

**Output:**

```
10 40 None
0
```

**Mutable Operations of dictionary**

**Adding an item in dictionary**

1. **dictionary-name[key]=value**

This syntax perform two operations

1. adding new item
2. modifying value of existing item

adding new item is done if key is not exists
modifying value of key is done, if key exists