1. Python | Ways to remove a key from dictionary.

In [1]:
```
test_dict = {"Arushi" : 22, "Anuradha" : 21, "Mani" : 21,  "Haritha" : 21}

print ("The dictionary before performing remove is : " + str(test_dict))

removed_value = test_dict.pop('Mani')

print ("The dictionary after remove is : " + str(test_dict))
print ("The removed key's value is : " + str(removed_value))

print ('\r')


removed_value = test_dict.pop('Manjeet', 'No Key found')

print ("The dictionary after remove is : " + str(test_dict))
print ("The removed key's value is : " + str(removed_value))
```

```
The dictionary before performing remove is : {'Arushi': 22, 'Anuradha': 21, 'Mani': 21, 'Haritha': 21}
The dictionary after remove is : {'Arushi': 22, 'Anuradha': 21, 'Haritha': 21}
The removed key's value is : 21

The dictionary after remove is : {'Arushi': 22, 'Anuradha': 21, 'Haritha': 21}
The removed key's value is : No Key found
```

2. Ways to sort list of dictionaries by values in Python – Using itemgetter.

In [2]:
```
from operator import itemgetter

my_list = [{ "name" : "Will", "age" : 56},
           { "name" : "Rob", "age" : 20 },
           { "name" : "Mark" , "age" : 34 },
           { "name" : "John" , "age" : 24 }]

print("The list sorted by age is : ")
print(sorted(my_list, key=itemgetter('age')))

print("The list sorted by age and name is : ")
print(sorted(my_list, key=itemgetter('age', 'name')))

print("The list sorted by age in descending order is : ")
print(sorted(my_list, key=itemgetter('age'),reverse = True))
```

```
The list sorted by age is :
[{'name': 'Rob', 'age': 20}, {'name': 'John', 'age': 24}, {'name': 'Mark', 'age': 34}, {'name': 'Will', 'age': 5
6}]
The list sorted by age and name is :
[{'name': 'Rob', 'age': 20}, {'name': 'John', 'age': 24}, {'name': 'Mark', 'age': 34}, {'name': 'Will', 'age': 5
6}]
The list sorted by age in descending order is :
[{'name': 'Will', 'age': 56}, {'name': 'Mark', 'age': 34}, {'name': 'John', 'age': 24}, {'name': 'Rob', 'age': 2
0}]
```

3. Ways to sort list of dictionaries by values in Python – Using lambda function

In [3]:
```
lis = [{ "name" : "Nandini", "age" : 20},
{ "name" : "Manjeet", "age" : 20 },
{ "name" : "Nikhil" , "age" : 19 }]

print ("The list printed sorting by age: ")
print (sorted(lis, key = lambda i: i['age']))

print ("\r")

print ("The list printed sorting by age and name: ")
print (sorted(lis, key = lambda i: (i['age'], i['name'])))

print ("\r")

print ("The list printed sorting by age in descending order: ")
print (sorted(lis, key = lambda i: i['age'],reverse=True))
```

```
The list printed sorting by age:
[{'name': 'Nikhil', 'age': 19}, {'name': 'Nandini', 'age': 20}, {'name': 'Manjeet', 'age': 20}]

The list printed sorting by age and name:
[{'name': 'Nikhil', 'age': 19}, {'name': 'Manjeet', 'age': 20}, {'name': 'Nandini', 'age': 20}]

The list printed sorting by age in descending order:
[{'name': 'Nandini', 'age': 20}, {'name': 'Manjeet', 'age': 20}, {'name': 'Nikhil', 'age': 19}]
```

4. Python | Merging two Dictionaries

In [6]:
```
dict_1 = {1: 'a', 2: 'b'}
dict_2 = {2: 'c', 4: 'd'}

print(dict_1 | dict_2)
```

```
{1: 'a', 2: 'c', 4: 'd'}
```

5. Python – Convert key-values list to flat dictionary

In [7]:
```
from itertools import product

test_dict = {'month' : [1, 2, 3],
             'name' : ['Jan', 'Feb', 'March']}

print("The original dictionary is : " + str(test_dict))

res = dict(zip(test_dict['month'], test_dict['name']))

print("Flattened dictionary : " + str(res))
```

```
The original dictionary is : {'month': [1, 2, 3], 'name': ['Jan', 'Feb', 'March']}
Flattened dictionary : {1: 'Jan', 2: 'Feb', 3: 'March'}
```

6. Python – Insertion at the beginning in OrderedDict

In [8]:
```
from collections import OrderedDict

iniordered_dict = OrderedDict([('akshat', '1'), ('nikhil', '2')])

iniordered_dict.update({'manjeet':'3'})
iniordered_dict.move_to_end('manjeet', last = False)

print ("Resultant Dictionary : "+str(iniordered_dict))
```

```
Resultant Dictionary : OrderedDict([('manjeet', '3'), ('akshat', '1'), ('nikhil', '2')])
```

7. Python | Check order of character in string using OrderedDict( )

In [9]:
```
from collections import OrderedDict

def checkOrder(input, pattern):

    dict = OrderedDict.fromkeys(input)

    ptrlen = 0
    for key,value in dict.items():
        if (key == pattern[ptrlen]):
            ptrlen = ptrlen + 1

        if (ptrlen == (len(pattern))):
            return 'true'

    return 'false'

if __name__ == "__main__":
    input = 'engineers rock'
    pattern = 'er'
    print (checkOrder(input,pattern))
```

```
true
```

8. Dictionary and counter in Python to find winner of election

In [10]:
```
from collections import Counter

def winner(input):

    votes = Counter(input)

    dict = {}

    for value in votes.values():

        dict[value] = []

    for (key,value) in votes.items():
        dict[value].append(key)

    maxVote = sorted(dict.keys(),reverse=True)[0]

    if len(dict[maxVote])>1:
        print (sorted(dict[maxVote])[0])
    else:
        print (dict[maxVote][0])

if __name__ == "__main__":
    input =['john','johnny','jackie','johnny',
            'john','jackie','jamie','jamie',
            'john','johnny','jamie','johnny',
            'john']
    winner(input)
```

```
john
```

9. Python – Append Dictionary Keys and Values (In order) in dictionary.

In [11]:
```
test_dict = {"Gfg" : 1, "is" :  3, "Best" : 2}

print("The original dictionary is : " + str(test_dict))

res = list(test_dict.keys()) + list(test_dict.values())

print("The ordered keys and values : " + str(res))
```

```
The original dictionary is : {'Gfg': 1, 'is': 3, 'Best': 2}
The ordered keys and values : ['Gfg', 'is', 'Best', 1, 3, 2]
```

10. Python | Sort Python Dictionaries by Key or Value

In [12]:
```
def dictionairy():

 key_value ={}

 key_value[2] = 56
 key_value[1] = 2
 key_value[5] = 12
 key_value[4] = 24
 key_value[6] = 18
 key_value[3] = 323

 print ("Task 1:-\n")
 print ("Keys are")

 for i in sorted (key_value.keys()) :
     print(i, end = " ")

def main():

    dictionairy()

if __name__=="__main__":
    main()
```

```
Task 1:-

Keys are
1 2 3 4 5 6
```

In [ ]:
```

```