## Aditya Kumar

## Python Dictionary Task

# 1. Write a Python function that takes in two dictionaries and returns a dictionary that contains only the keys that are present in both dictionaries.

In [1]:

```
def fun(dict1,dict2):
    dict3={}
    for key in dict1.keys():
        if key in dict2.keys(): # using membership opertaor "in" or "not in"
            dict3[key]=dict1[key]

            #dict3[key]=[dict1[key],dict2[key]]

    return dict3
```

In [2]:

```
dict1={"name":"aditya","roll":"304","branch":"ECE"}
dict2={"name":"Raju","subject":"CSE",'branch':"It"}

fun(dict1,dict2)
```

Out[2]:

```
{'name': 'aditya', 'branch': 'ECE'}
```

***Another Approach***

In [3]:

```
def fun(dict1,dict2):
    dict3={}
    for key in dict1.keys():
        if key in dict2.keys(): # using membership opertaor "in" or "not in"
            #dict3[key]=dict1[key]

            dict3[key]=[dict1[key],dict2[key]] # it will take list of values.

    return dict3
```

In [4]:

```
dict1={"name":"aditya","roll":"304","branch":"ECE"}
dict2={"name":"Raju","subject":"CSE",'branch':"It"}

fun(dict1,dict2)
```

Out[4]:

```
{'name': ['aditya', 'Raju'], 'branch': ['ECE', 'It']}
```

In [5]:

```
# Python code to creat a single user defined dictionary.

# key=[]
# dict4={}
# for i in range(int(input("How many key you want to add:"))):
#     key.append(input("Enter {} key name:".format(i+1)))
#     dict4[key[i]]=input("Enter {} key value:".format(i+1))

# dict4
```

In [6]:

```
# python code to crate a user defined dictionary list.

# def program1():
#     key=[]
#     diclist=[]
#     dictnew={}
#     for j in range(int(input("Enter the no. of dict:"))):
#         for i in range(int(input("How many items you want in each dictionar
#             key.append(input("enter the {} key name :".format(i+1)))
#             dictnew[key[i]]=input("enter the {} key value:".format(i+1))
#             diclist.append(dictnew)
#     return diclist
# progam1()
```

## 2. Write a Python function that takes in a list of dictionaries and a key, and returns a new list of dictionaries sorted by the value of the specified key in each dictionary.

In [7]:

```
# #sort string data type in list
# l=['12','4','10','45','23','a','d','c']
# l.sort(key=str)
# print(l)

# # sort integer data type in list
# l=[1,5,2,10,6,7,3]
# l.sort(key=int)
# l
```

In [8]:

```
dict1={"Age":22,"Marks":85,"CSE":20,'Science':75}
dict2={"Age":34,"Marks":685,"CSE":220,'Science':15}
dict3={"Age":62,"Marks":89,"CSE":90,'Science':65}
mylist=[dict1,dict2,dict3]

# defined function
def fun(mylist,key1):
    l=[]
    newdictlist=[]
    for i in range(len(mylist)):
        if key1 in mylist[i].keys():
            l.append(mylist[i][key1])
        else:
            pass
    l.sort()

    for i in l:
        for j in mylist:
            if i in j.values():
                newdictlist.append(j)

    return newdictlist

# calling fucntion
key3='CSE'
fun(mylist,key3)
```

Out[8]:

```
[{'Age': 22, 'Marks': 85, 'CSE': 20, 'Science': 75},
 {'Age': 62, 'Marks': 89, 'CSE': 90, 'Science': 65},
 {'Age': 34, 'Marks': 685, 'CSE': 220, 'Science': 15}]
```

**Another Method**

- Syntax
    - sorted(iterable, key=key, reverse=Ture/False)
    - where key is optional , by default key is none.
    - where reverse is alos optional , by default it is in ascending order.

In [9]:

```
def fun(mylist,key):
    def get_value(item):
        return item[key]
    return sorted(mylist,key=get_value)

key3='Marks'
fun(mylist,key3)
```

Out[9]:

```
[{'Age': 22, 'Marks': 85, 'CSE': 20, 'Science': 75},
 {'Age': 62, 'Marks': 89, 'CSE': 90, 'Science': 65},
 {'Age': 34, 'Marks': 685, 'CSE': 220, 'Science': 15}]
```

In [11]:

```
# m=[2,1,23,12,0,100,'Age', 'Marks', 'CSE', 'Science',5]
# print(m)

# mod=[]
# modstr=[]
# for i in range(len(m)):
#     if type(m[i])!=str:
#         mod.append(m[i])
#         mod.sort()
#     else:
#         modstr.append(m[i])
#         print(modstr)
#         modstr.sort()
# print(m)
# print(mod)
# print(modstr)
# print(mod+modstr)
```

## 3. Write a Python function that takes in a dictionary and returns the key with the maximum value.

In [12]:

```
def fun(dict1):
    newlist=[]
    newdict={}

    for key in dict1.keys():
        newlist.append(dict1[key])
    maximum=max(newlist)

    for key in dict1.keys():
        if dict1[key]==maximum:
            newdict[key]=maximum
        else:
            pass

    x="updated new Dictionary : {} and Key with maximum Value : {}".format(ne
    return x
```

In [13]:

```
dict12={"age":22,"Marks":85,"Math":20,'Science':75}

# calling function
fun(dict12)
```

Out[13]:

```
"updated new Dictionary : {'Marks': 85} and Key with maximum Value : dict_ke
ys(['Marks'])"
```

In [14]:

```
# # In case of string and integer values we will use this method to find the r

# lis=[1,2,5,3,7,4,'a','b']
# b=[]
# print("Initial List :",lis)
# for i in range(len(lis)):
#     try:
#         if type(lis[i])!=str:
#             b.append(lis[i])
# #             print(b)
#     except:
#         b.remove(lis[i])
# print("Updated List :",b)

# # Now after that follow the previous cell method to find the key with maximu
```

## 4. Write a Python function that takes in a dictionary and returns a new dictionary that contains only the keys that have even values.

In [15]:

```
def fun(dict1):
    dictnew={}
    for key in dict1.keys():
        try:
            if int(dict1[key]) % 2 == 0:
                dictnew[key]=dict1[key]
        except:
            pass

    return dictnew
```

In [16]:

```
dict1={'name':'Aditya',"age":22,"gender":"Male","Marks":85,"Math":20}

# calling fucntion
fun(dict1)
```

Out[16]:

```
{'age': 22, 'Math': 20}
```

## 5. Write a Python function that takes in a dictionary and returns a new dictionary that contains the keys sorted by their values in descending order.

In [17]:

```
dict1={"Age":22,"Marks":85,"CSE":20,'Science':75}

# defining the function
def fun(dict1):
    valuelist=[]
    newdict={}
    for value in dict1.values():
        valuelist.append(value)
    valuelist.sort(reverse=True)

    #return valuelist

    for i in valuelist:
        for key in dict1.keys():
            if dict1[key]==i:
                 newdict[key]=i

    return newdict

#calling function
fun(dict1)
```

Out[17]:

```
{'Marks': 85, 'Science': 75, 'Age': 22, 'CSE': 20}
```

#### Another Method

In [18]:

```
def sort_dict_by_value(dictionary):
    sorted_keys = sorted(dictionary, key=dictionary.get, reverse=True)
    sorted_dict = {key: dictionary[key] for key in sorted_keys}
    return sorted_dict
```

In [19]:

```
dict1={"Age":22,"Marks":85,"CSE":20,'Science':75}
print(dict1.get)

sort_dict_by_value(dict1)
```

```
<built-in method get of dict object at 0x000001677B41FEC0>
```

Out[19]:

```
{'Marks': 85, 'Science': 75, 'Age': 22, 'CSE': 20}
```

## 6. Write a Python function that takes in a list of dictionaries and returns a new list of dictionaries that contain only the keys that are present in all dictionaries.

In [20]:

```
dict1={"Age":22,"Marks":85,"CSE":20,'Science':75}
dict2={"Marks":685,"CSE":220,'Science':15}
dict3={"Age":62,"Marks":89,"CSE":90,'Science':65}
mylist=[dict1,dict2,dict3]

# defining fucntion
def fun(mylist):
    a={} # empty set to contain keys
    b=[] # empty list of set of keys
    newdict={} # empyt new dictionary
    for i in mylist:
        a=set(i.keys())
        b.append(a)
    a=set.intersection(*[keyset for keyset in b])
    for i in a:
        newdict[i]=dict1[i]
    return newdict

fun(mylist)
```

Out[20]:

```
{'Marks': 85, 'CSE': 20, 'Science': 75}
```

## 7. Write a Python function that takes in a dictionary and a list of keys, and returns a new dictionary that contains only the specified keys and their corresponding values.

In [21]:

```
def fun(dict1,mylist):
    newdict={}
    for key in dict1.keys():
        if key in mylist:
            newdict[key]=dict1[key]
        else:
            pass

    return newdict
```

In [22]:

```
dict1={'name':'Aditya',"age":22,"gender":"Male","Marks":85,"Math":20}
mylist=['name','computer','science','gender']

# calling function
fun(dict1,mylist)
```

Out[22]:

```
{'name': 'Aditya', 'gender': 'Male'}
```

## 8. Write a Python function that takes in a dictionary and a list of keys, and returns a new dictionary that contains only the keys that are not present in the list.

In [23]:

```
def fun(dict1,mylist):
    newdict={}
    for key in dict1.keys():
        if key in mylist:
            pass
        else:
            newdict[key]=dict1[key]
    return newdict
```

In [24]:

```
dict1={'name':'Aditya',"age":22,"gender":"Male","Marks":85,"Math":20}
mylist=['name','computer','science','gender']

#calling function
fun(dict1,mylist)
```

Out[24]:

```
{'age': 22, 'Marks': 85, 'Math': 20}
```

## 9. Write a Python function that takes in a list of dictionaries and a key, and returns a new list of dictionaries that contain only the dictionaries with the specified key.

In [25]:

```
def fun(mylist,key):
    newdictList=[]
    for i in mylist:
        if key in i.keys():
            newdictList.append(i)
        else:
            pass
    return newdictList
```

In [26]:

```
dict1={"Age":22,"Marks":85,"CSE":20,'Science':75}
dict2={"Marks":685,"CSE":220,'Science':15}
dict3={"Age":62,"Marks":89,"CSE":90,'Science':65}

mylist=[dict1,dict2,dict3]
key ="Age"

#calling fucntion
fun(mylist,key)
```

Out[26]:

```
[{'Age': 22, 'Marks': 85, 'CSE': 20, 'Science': 75},
 {'Age': 62, 'Marks': 89, 'CSE': 90, 'Science': 65}]
```

# 10. Write a Python function that takes in a dictionary and a list of keys, and returns a new dictionary that contains only the keys that are present in the dictionary and not in the list

In [27]:

```python
def fun(dict1,mylist):
    newdict={}
    for key in dict1.keys():
        if key in mylist:
            pass
        else:
            newdict[key]=dict1[key]
    return newdict
```

In [28]:

```python
dict1={'name':'Aditya',"age":22,"gender":"Male","Marks":85,"Math":20}
mylist=['name','computer','science','gender']

#calling function
fun(dict1,mylist)
```

Out[28]:

```
{'age': 22, 'Marks': 85, 'Math': 20}
```

In [ ]:

```
```

In [ ]:

```
```