In [2]:
```
# # Write a Python script to concatenate the following dictionaries to create a new one.
# # Sample Dictionary :
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
# Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

In [4]:
```
dic={}
for x in (dic1,dic2,dic3):
    dic.update(x)
print(dic)
```

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

In [7]:
```
# Write a Python script to check whether a given key already exists in a dictionary.
if 5 in dic:
    print("key is present")
else:
    print("not")
```

```
key is present
```

In [11]:
```
# Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and
# the values are the square of the keys.

def new(d):
    new_dic={}
    for x in range(1,d):
        new_dic.update({x:x**2})
    print(new_dic)
```

In [12]:
```
new(30)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15:
225, 16: 256, 17: 289, 18: 324, 19: 361, 20: 400, 21: 441, 22: 484, 23: 529, 24: 576, 25: 625, 26: 676, 27: 72
9, 28: 784, 29: 841}
```

In [1]:
```
# Write a Python script to merge two Python dictionaries.
d1={"name1":"Ritu","city1":"Pune"}
d2={"name2":"Seeta","city2":"Banglore"}
d1.update(d2)
print(d1)
```

```
{'name1': 'Ritu', 'city1': 'Pune', 'name2': 'Seeta', 'city2': 'Banglore'}
```

In [2]:
```
def merge_dict(*dic):
    d={}
    for x in dic:
        d.update(x)
    print(d)
```

In [4]:
```
d1={"name1":"Ritu","city1":"Pune"}
d2={"name2":"Seeta","city2":"Banglore"}
merge_dict(d1,d2)
```

```
{'name1': 'Ritu', 'city1': 'Pune', 'name2': 'Seeta', 'city2': 'Banglore'}
```

In [8]:
```
# Create a dictionary 'd' with color names as keys and corresponding numerical values as values.
d = {'Red': 1, 'Green': 2, 'Blue': 3}
for color_key, value in d.items():
    print(color_key, 'corresponds to ', d[color_key])
```

```
Red corresponds to  1
Green corresponds to  2
Blue corresponds to  3
```

In [10]:
```
for color_key in d.keys():
    print(color_key, 'corresponds to ', d[color_key])
```

```
Red corresponds to  1
Green corresponds to  2
Blue corresponds to  3
```

In [11]:
```
# Write a Python program to sum all the items in a dictionary.
d={1:200,2:400,3:500}
sum=0
for value in d.values():
    sum+=value
print(sum)
```

```
1100
```

In [13]:
```
# Write a Python program to multiply all the items in a dictionary.
d={1:200,2:400,3:500}
prod=1
for value in d.values():
    prod*=value
print(prod)
```

```
40000000
```

In [17]:
```
#  Write a Python program to get the maximum and minimum values of a dictionary.
d={1:200,2:400,3:500}
maxi=max(d.values())
mini=min(d.values())
print(maxi)
print(mini)
```

```
500
200
```

In [32]:
```
# Write a Python program to sort a given dictionary by key and values
d={1:200,2:400,3:500,5:300,4:100}
print(sorted(d.keys()))
print(sorted(d.values()))
```

```
[1, 2, 3, 4, 5]
[100, 200, 300, 400, 500]
```

In [33]:
```
print(dict(sorted(d.items())))
```

```
{1: 200, 2: 400, 3: 500, 4: 100, 5: 300}
```

In [20]:
```
#  Write a Python program to map two lists into a dictionary.
keys=["name","age"]
values=["July",23]
d=dict(zip(keys,values))
print(d)
```

```
{'name': 'July', 'age': 23}
```

In [ ]:
```
#  Write a Python program to remove duplicates from the dictionary.
```

In [34]:
```
details={"stud1":{"name":"Riya","age":24,"city":"Pune","education":"MSC","grade":8.9},
         "stud2":{"name":"Siya","age":20,"city":"nashik","education":"post-graduate", "grade":8.5},
        "stud3":{"name":"Vaibhav","age":18,"city":"nagar","education":"graduation","grade":7.8},
        "stud4":{"name":"Riya","age":24,"city":"Pune","education":"MSC","grade":8.9}}
print(details)
```

```
{'stud1': {'name': 'Riya', 'age': 24, 'city': 'Pune', 'education': 'MSC', 'grade': 8.9}, 'stud2': {'name': 'Si
ya', 'age': 20, 'city': 'nashik', 'education': 'post-graduate', 'grade': 8.5}, 'stud3': {'name': 'Vaibhav', 'a
ge': 18, 'city': 'nagar', 'education': 'graduation', 'grade': 7.8}, 'stud4': {'name': 'Riya', 'age': 24, 'cit
y': 'Pune', 'education': 'MSC', 'grade': 8.9}}
```

In [37]:
```
det={}
for key, value in details.items():
    if value not in det.values():
        det.update({key:value})
print(det)
```

```
{'stud1': {'name': 'Riya', 'age': 24, 'city': 'Pune', 'education': 'MSC', 'grade': 8.9}, 'stud2': {'name': 'Si
ya', 'age': 20, 'city': 'nashik', 'education': 'post-graduate', 'grade': 8.5}, 'stud3': {'name': 'Vaibhav', 'a
ge': 18, 'city': 'nagar', 'education': 'graduation', 'grade': 7.8}}
```

In [ ]:
```
# Write a Python program to check if a dictionary is empty or not.
```

In [3]:
```
d={}
print(len(d))
```

```
0
```

In [4]:
```
if len(d)==0:
    print("empty dictionary")
else:
    print("not empty")
```

```
empty dictionary
```

In [5]:
```
def empty_dict(d):
    if len(d)==0:
        print("empty dictionary")
    else:
        print("not empty")
    return d
```

In [6]:
```
empty_dict({1:23})
```

```
not empty
```

Out[6]:
```
{1: 23}
```

In [7]:
```
empty_dict({})
```

```
empty dictionary
```

Out[7]:
```
{}
```

```
Write a Python program to print all distinct values in a dictionary.
Sample Data : [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"},
{"VIII":"S007"}]
Expected Output : Unique Values: {'S005', 'S002', 'S007', 'S001', 'S009'}
```

In [83]:
```
data=[{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"},{"VIII":"S007"}
```

In [85]:
```
d=[]
for x in data:
    for i in x.values():
        if i not in d:
            d.append(i)
print(d)
```

```
['S001', 'S002', 'S005', 'S009', 'S007']
```

In [88]:
```
d1=set()
for x in data:
    for i in x.values():
        d1.add(i)
print(d1)
```

```
{'S001', 'S007', 'S005', 'S002', 'S009'}
```

```
Write a Python program to create a dictionary from a string.
Sample string : 'w3resource'
```

```
Expected output: {'w': 1, '3': 1, 'r': 2, 'e': 2, 's': 1, 'o': 1, 'u': 1, 'c': 1}
```

In [90]:
```
s='w3resource'
d={}
for x in s:
    if x not in d:
        d.update({x:s.count(x)})
print(d)
```

```
{'w': 1, '3': 1, 'r': 2, 'e': 2, 's': 1, 'o': 1, 'u': 1, 'c': 1}
```

```
Write a Python program to get the top three items in a shop.
Sample data: {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}
Expected Output:
item4 55
item1 45.5
item3 41.3
```

In [96]:
```
item = {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}
list1=sorted(items.values(),reverse=True)[:3]
for l in list1:
    for i ,j in item.items():
        if l==j:
            print(i,j)
```

```
item4 55
item1 45.5
item3 41.3
```

```
Write a Python program to sort by value.
Sample data : {'Math':81, 'Physics':83, 'Chemistry':87}
Expected data: [('Chemistry', 87), ('Physics', 83), ('Math', 81)]
```

In [103]:
```
data={'Math':81, 'Physics':83, 'Chemistry':87}
list1=sorted(data.values(),reverse=True)
d={}
for l in list1:
    for i ,j in data.items():
        if l==j:
            d.update({i:j})
print(d)
```

```
{'Chemistry': 87, 'Physics': 83, 'Math': 81}
```

In [ ]:
```

```

In [104]:
```
# Write a Python program to create a dictionary grouping a sequence of key-value pairs into a dictionary of lis
# Original list:
d=[('yellow', 1), ('blue', 2), ('yellow', 3), ('blue', 4), ('red', 1)]
# Grouping a sequence of key-value pairs into a dictionary of lists:
# {'yellow': [1, 3], 'blue': [2, 4], 'red': [1]}
```

In [125]:
```
d1={}
for x in d:
    if x[0] not in d1.keys():
            d1.update({x[0]:[x[1]]})
    elif x[0] in d1.keys():
        d1[x[0]].append(x[1])
print(d1)
```

```
{'yellow': [1, 3], 'blue': [2, 4], 'red': [1]}
```

```
52. Write a Python program to extract a list of values from a given list of dictionaries.
Original Dictionary:
[{'Math': 90, 'Science': 92}, {'Math': 89, 'Science': 94}, {'Math': 92, 'Science': 88}]
Extract a list of values from said list of dictionaries where subject = Science
[92, 94, 88]
Original Dictionary:
[{'Math': 90, 'Science': 92}, {'Math': 89, 'Science': 94}, {'Math': 92, 'Science': 88}]
Extract a list of values from said list of dictionaries where subject = Math
```

```
[90, 89, 92]
```

In [14]:
```
l=[{'Math': 90, 'Science': 92}, {'Math': 89, 'Science': 94}, {'Math': 92, 'Science': 88}]
def sub(n):
    l1=[]
    for x in l:
        for k,v in x.items():
            if k==n:
                l1.append(v)
    return l1
```

In [15]:
```
sub("Math")
```

Out[15]:
```
[90, 89, 92]
```

In [17]:
```
sub("Science")
```

Out[17]:
```
[92, 94, 88]
```

```
Write a Python program to convert a dictionary into a list of lists.
Original Dictionary:
{1: 'red', 2: 'green', 3: 'black', 4: 'white', 5: 'black'}
Convert the said dictionary into a list of lists:
[[1, 'red'], [2, 'green'], [3, 'black'], [4, 'white'], [5, 'black']]
Original Dictionary:
{'1': 'Austin Little', '2': 'Natasha Howard', '3': 'Alfred Mullins', '4': 'Jamie Rowe'}
Convert the said dictionary into a list of lists:
[['1', 'Austin Little'], ['2', 'Natasha Howard'], ['3', 'Alfred Mullins'], ['4', 'Jamie Rowe']]
```

In [24]:
```
def dl(d):
    list1=[]
    for k, v in d.items():
        list1.append([k,v])
    return list1
```

In [25]:
```
d= {1: 'red', 2: 'green', 3: 'black', 4: 'white', 5: 'black'}
dl(d)
```

Out[25]:
```
[[1, 'red'], [2, 'green'], [3, 'black'], [4, 'white'], [5, 'black']]
```

In [26]:
```
d1={'1': 'Austin Little', '2': 'Natasha Howard', '3': 'Alfred Mullins', '4': 'Jamie Rowe'}
dl(d1)
```

Out[26]:
```
[['1', 'Austin Little'],
 ['2', 'Natasha Howard'],
 ['3', 'Alfred Mullins'],
 ['4', 'Jamie Rowe']]
```

```
 Write a Python program to filter even numbers from a dictionary of values.
Original Dictionary:
{'V': [1, 4, 6, 10], 'VI': [1, 4, 12], 'VII': [1, 3, 8]}
Filter even numbers from said dictionary values:
{'V': [4, 6, 10], 'VI': [4, 12], 'VII': [8]}
Original Dictionary:
{'V': [1, 3, 5], 'VI': [1, 5], 'VII': [2, 7, 9]}
Filter even numbers from said dictionary values:
{'V': [], 'VI': [], 'VII': [2]}
```

In [28]:
```
def even_value(d):
    ed={}
    for k, v in d.items():
        n=[x for x in v if x%2==0]
        ed[k]=n
    print(ed)
```

In [29]:
```
d={'V': [1, 4, 6, 10], 'VI': [1, 4, 12], 'VII': [1, 3, 8]}
even_value(d)
```

```
{'V': [4, 6, 10], 'VI': [4, 12], 'VII': [8]}
```

In [30]:
```
d1={'V': [1, 3, 5], 'VI': [1, 5], 'VII': [2, 7, 9]}
even_value(d1)
```

```
{'V': [], 'VI': [], 'VII': [2]}
```

```
 Write a Python program to find the shortest list of values for the keys in a given dictionary.
Original Dictionary: {'V': [10, 12], 'VI': [10], 'VII': [10, 20, 30, 40], 'VIII': [20], 'IX': [10, 30, 50,
70], 'X': [80]} Shortest list of values with the keys of the said dictionary: ['VI', 'VIII', 'X']
```

In [41]:
```
def short(d):
    kl=[]
    for k,v in d.items():
        if len(v)<=1:
            kl.append(k)
    print(kl)
```

In [42]:
```
d={'V': [10, 12], 'VI': [10], 'VII': [10, 20, 30, 40], 'VIII': [20], 'IX': [10, 30, 50, 70], 'X': [80]}
short(d)
```

```
['VI', 'VIII', 'X']
```

```
Write a Python program to count the frequency of a dictionary.
Original Dictionary:
{'V': 10, 'VI': 10, 'VII': 40, 'VIII': 20, 'IX': 70, 'X': 80, 'XI': 40, 'XII': 20}
Count the frequency of the said dictionary:
Counter({10: 2, 40: 2, 20: 2, 70: 1, 80: 1})
```

In [49]:
```
def val_freq(d):
    freq_dict={}
    count=0
    for v in d.values():
        if v in freq_dict:
            count+=1
            freq_dict[v]=count
        else:
            count=1
            freq_dict[v]=count
    print(freq_dict)
```

In [50]:
```
d={'V': 10, 'VI': 10, 'VII': 40, 'VIII': 20, 'IX': 70, 'X': 80, 'XI': 40, 'XII': 20}
val_freq(d)
```

```
{10: 2, 40: 2, 20: 3, 70: 1, 80: 1}
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```
# write a python programm to check key values values
Function to get the max element from a list.
Function to check whether a string is palindrome or not.
-         Find the frequency of each distinct element in the list using a Python dictionary.
Input:  [1, 1, 1, 5, 5, 3, 1, 3, 3, 1, 4, 4, 4, 2, 2, 2, 2]
```

Output: {1:5, 2:4, 3:3, 4:3, 5:2}

In [9]:
```
Input=[1, 1, 1, 5, 5, 3, 1, 3, 3, 1, 4, 4, 4, 2, 2, 2, 2]
out={}
def freq(Input):
    freq=0
    for x in Input:
        if x in out.keys():
            out.values=freq+1
        else:
            freq=1
print(out)
```

{}

In [ ]: