

## Shubhangi\_Dhikale\_35

### 1. Python program to sort Python Dictionaries by Keys

```
In [3]: sub_marks={"phy":95,"chem":90,"math":97,"Eng":93,"Bio":93}
a=sorted(sub_marks.items())
print(dict(a))

{'Bio': 93, 'Eng': 93, 'chem': 90, 'math': 97, 'phy': 95}
```

### 2. Python program to sort Python Dictionaries by Value

```
In [2]: sub_marks={"phy":95,"chem":90,"math":97,"Eng":93,"Bio":91}
sorted_dict={}
sorted_values=sorted(sub_marks.values())
for i in sorted_values:
    for j in sub_marks.keys():
        if sub_marks[j]==i:
            sorted_dict[j]=sub_marks[j]
print(sorted_dict)

{'chem': 90, 'Bio': 91, 'Eng': 93, 'phy': 95, 'math': 97}
```

### 3. Python program to find the sum of all items in a dictionary

```
In [1]: sub_marks={"phy":95,"chem":90,"math":97,"Eng":93,"Bio":91}
print(sum(sub_marks.values()))

466
```

### 4. Python program to remove a key from a dictionary

```
In [6]: sub_marks={"phy":95,"chem":90,"math":97,"Eng":93,"Bio":91}
sub_marks.pop("phy")
sub_marks

Out[6]: {'chem': 90, 'math': 97, 'Eng': 93, 'Bio': 91}
```

### 5. Python program to merge two Dictionaries

```
In [8]: sub_marks={"phy":95,"chem":90,"math":97,"Eng":93}
student_id={"A":2,"B":3,"C":5,"D":7}
sub_marks.update(student_id)
sub_marks
```

```
Out[8]: {'phy': 95, 'chem': 90, 'math': 97, 'Eng': 93, 'A': 2, 'B': 3, 'C': 5, 'D': 7}
```

## 6.Program to create grade calculator in Python

```
In [11]: maths=94
phy=89
chem=90
bio=98
python=96

def sub_grades(maths,chem,phy,bio,python):
    average=((maths+chem+phy+bio+python)/500)*100
    return average
def grade_cal(average):
    grade=""
    if average>=90:
        print("A+ grade")
    elif average>=80 and average<90:
        print("A grade")
    elif average >=70 and average <80:
        print("B grade")
    else:
        print("C grade")
    return grade
average=sub_grades(maths,chem,phy,bio,python)
print("percentage:",average)
print(grade_cal(average))
```

```
percentage: 93.4
A+ grade
```

## 7.Print anagrams together in Python using List and Dictionary

```
In [25]: anagram=["act","silent","cat","listen","ok","dog","god"]
dict1={}
for i in anagram:
    a=" ".join(sorted(i))
    if a in dict1.keys():
        dict1[a].append(i)
    else:
        dict1[a]=[]
        dict1[a].append(i)
result=" "
for i,val in dict1.items():
    result=result+" ".join(val)+" "
print(result)

act cat silent listen ok dog god
```

## 8.Check if binary representations of two numbers are an anagram

```
In [12]: n1=int(input("Enter the 1st number:"))
n2=int(input("Enter the 2nd number:"))
bin1=bin(n1)
bin2=bin(n2)
if bin1 !=bin2 and (sorted(bin1)==sorted(bin2)):
    print(f'{bin1} and {bin2} are an anagram')
else:
    print(f'{bin1} and {bin2} are not an anagram')

Enter the 1st number:2
Enter the 2nd number:3
0b10 and 0b11 are not an anagram
```

## 9.Python Counter to find the size of the largest subset of anagram words

```
In [27]: a="act cat cat listen silent dog god god ogd"
b=a.split(" ")
dict1={}
max1=0
for i in range(0,len(b)):
    b[i]="".join(sorted(b[i]))
    dict1[b[i]]=b.count(b[i])
print(dict1)

{'act': 3, 'eilnst': 2, 'dgo': 4}
```

## 10.Python Dictionary to find mirror characters in a string

```
In [13]: l1=["cat","rat","mat"]
print('char:',l1)
l2=[]
result={}
for i in l1:
    a=i[::-1]
    l2.append(a)
print("mirror char:",l2)

for index,l1 in enumerate(l1):
    result[l1]=l2[index]
print("dict to get mirror char:",result)
```

```
char: ['cat', 'rat', 'mat']
mirror char: ['tac', 'tar', 'tam']
dict to get mirror char: {'cat': 'tac', 'rat': 'tar', 'mat': 'tam'}
```

## 11.Counting the frequencies in a list using a dictionary in Python

```
In [15]: string="We are Learning python And Data science and Machine learning in Python da
d={}
str1=string.lower().split()
for i in str1:
    d[i]=str1.count(i)
print(d)
```

```
{'we': 1, 'are': 1, 'learning': 2, 'python': 2, 'and': 2, 'data': 2, 'science':
2, 'machine': 1, 'in': 1}
```

## 12.Python program to convert a list of Tuples into Dictionary

```
In [17]: l1=[("a",1),("b",3),("c",5),("d",6)]
print("list of tuple is:",l1)
def my_fun(l1):
    dictionary=dict(l1)
print("list of tuple into dictionary:",dictionary)
```

```
list of tuple is: [('a', 1), ('b', 3), ('c', 5), ('d', 6)]
list of tuple into dictionary: {'a': 1, 'b': 3, 'c': 5, 'd': 6}
```

## 13.Scraping And Finding Ordered Words In A Dictionary using Python

```
In [ ]:
```

## 14.Create a list of tuples from the given list having a number and its cube in each tuple

```
In [18]: list1=[2,3,4,5,6]
print(list1)
my_result=[(i,i**3) for i in list1]
print("the result is:",my_result)
```

```
[2, 3, 4, 5, 6]
the result is: [(2, 8), (3, 27), (4, 64), (5, 125), (6, 216)]
```

## 15. a list of tuples by the second item

```
In [19]: list1=[("a",1),("b",2),("c",3),("d",4)]
print("original list:",list1)
length=len(list1)
for i in range(length):
    for j in range(length-i-1):
        if (list1[j][1]>list1[j+1][1]):
            temp=list1[j+1]
            list1[j]=list1[j+1]
            list1[j+1]=temp
print("sorted list1:",list1)
```

```
original list: [('a', 1), ('b', 2), ('c', 3), ('d', 4)]
sorted list1: [('a', 1), ('b', 2), ('c', 3), ('d', 4)]
```

## 16.Python Program for Insertion Sort

```
In [21]: def insertionsort(my_list):
    for i in range(1,len(my_list)):
        current_element=my_list[i]
        pos=i
        while current_element<my_list[pos-1] and pos>0:
            my_list[pos]=my_list[pos-1]
            pos=pos-1
        my_list[pos]=current_element
list1=[2,4,3,5,1]
print("unsorted list:",list1)

insertionsort(list1)
print("insertion sort list:",list1)
```

```
unsorted list: [2, 4, 3, 5, 1]
insertion sort list: [1, 2, 3, 4, 5]
```

## 17.Python Program for SelectionSort

```
In [22]: list1=[56,52,42,85,46]
print("unsorted list:",list1)

for i in range(len(list1)):
    min_i=min(list1[i:])
    min_ind=list1.index(min_i)
    list1[i],list1[min_ind]=list1[min_ind],list1[i]
print("selection sort list:",list1)
```

unsorted list: [56, 52, 42, 85, 46]  
selection sort list: [42, 46, 52, 56, 85]

## 18.Python Program for Bubble Sort

```
In [23]: list1=[56,52,42,85,46]
print("unsorted list:",list1)

for j in range(len(list1)-1):
    for i in range(len(list1)-1):
        if list1[i]>list1[i+1]:
            list1[i],list1[i+1]=list1[i+1],list1[i]

print("bubble sort list:",list1)
```

unsorted list: [56, 52, 42, 85, 46]  
bubble sort list: [42, 46, 52, 56, 85]

## 19.Python Program for Merge Sort

In [ ]:

## 20.Python Program for QuickSortSort

In [ ]: