Python Programming - Lab - 7

March 11, 2025

Python Programming - 2301CS404

Lab - 7

OM BHUT | 23010101033 | 122

1 Set & Dictionary

1.0.1 01) WAP to iterate over a set.

[ ]:
```
s = {1,2,3,4,5,6,7}
for i in s:
    print(i,end=" ")
```

1.0.2 02) WAP to convert set into list, string and tuple.

[ ]:
```
s = {1,2,3,4,5,6,7}
l1 = list(s)
t1 = tuple(s)
s1 = ""
for i in s:
    s1 += str(i) + " "
print(l1)
print(t1)
print(s1)
```

1.0.3 03) WAP to find Maximum and Minimum from a set.

[ ]:
```
s = {1,2,3,4,5,6,7}
print(max(s))
print(min(s))
```

1.0.4 04) WAP to perform union of two sets.

[ ]:
```
s1 = {1,2,3,4,5,6,7}
s2 = {4,5,6,7,8,9,10}
s3 = s1.union(s2)
s3
```

1.0.5 05) WAP to check if two lists have at-least one element common.

[ ]:
```
s1 = {1,2,3,4,5,6,7}
s2 = {4,5,6,7,8,9,10}
s3 = s1.intersection(s2)
print("yes" if len(s3) >= 1 else "no")
```

1.0.6 06) WAP to remove duplicates from list.

[ ]:
```
l1 = [1,1,1,1,2,2,2,3,3,3]
print(list(set(l1)))
```

1.0.7 07) WAP to find unique words in the given string.

[ ]:
```
s = "hello hello hi bye bye".split(" ")
s1 = " ".join(set(s))
s1
```

1.0.8 08) WAP to remove common elements of set A & B from set A.

[ ]:
```
s1 = {1,2,3,4,5,6,7}
s2 = {4,5,6,7,8,9,10}
s3 = s1 - s2
s3
```

1.0.9 09) WAP to check whether two given strings are anagram or not using dictionary.

[13]:
```
def checkAnagram(s1,s2):
    return sorted(s1) == sorted(s2)

def checkAnagramUsingDictionary(s1,s2):
    d1 = {i:s1.count(i) for i in s1}
    d2 = {i:s2.count(i) for i in s1}
    return d1 == d2

s1 = "are you why"
s2 = "why are you"
print(checkAnagramUsingDictionary(s1,s2))
```

```
True
```

1.0.10 10) WAP to find common elements in three lists using set.

[20]:
```
l1=[1,2,3,5]
l2=[1,3,9]
l3=[1,5,7,3]
```

```
s1 = set(l1)
s2 = set(l2)
s3 = set(l3)
print(s1.intersection(s2,s3))
```

```
{1, 3}
```

[21]:
```
s1 = set(s1)
s1.intersection(l2,l3)
```

[21]:
```
{1, 3}
```

1.0.11 11) WAP to count number of vowels in given string using set.

[22]:
```
def countVowels(set1,str1):
    count = 0
    for i in str1:
        if i in set1:
            count+=1
    return count
s1 = {'a','e','i','o','u'}
print(countVowels(s1,"aaabbdfgdu"))
```

```
4
```

1.0.12 12) WAP to check if a given string is binary string or not.

[3]:
```
set1 = {'0','1'}
str1 = "010111010"

count1 = 0
for i in str1:
    if i in set1:
        count1+=1
print(count1==len(str1))
```

```
True
```

1.0.13 13) WAP to sort dictionary by key or value.

[10]:
```
d1 = {3:"bhavya",2:"avi",4:"chaman"}

print(sorted(d1.values()))
print(sorted(d1.keys()))
```

```
['avi', 'bhavya', 'chaman']
[2, 3, 4]
```

1.0.14 14) WAP to find the sum of all items (values) in a dictionary given by user. (Assume: values are numeric)

[11]:
```
d1 = {}
sum = 0
for i in range(5):
    key = input("enter key")
    value = int(input("enter value"))
    sum+=value
    d1[key] = value
print(sum)
```

```
21
```

1.0.15 15) WAP to handle missing keys in dictionaries.

Example : Given, dict1 = {'a': 5, 'c': 8, 'e': 2}

if you look for key = 'd', the message given should be 'Key Not Found', otherwise print the value of 'd' in dict1.

[19]:
```
dict1 = {'a': 5, 'c': 8, 'e': 2}
userInput = 'a'
valueOfKey = dict1.get(userInput)
if (valueOfKey == None):
    print("Key Not Found")
else:
    print(dict1[userInput])
```

```
5
```