

* Set function eliminates duplicates

Set(x)

Split(x) → is used to split the word & Sentence.

WAP accept a string and display frequency of characters.

x = input("Enter a string: ")

d1 = {}

for i in set(x):

n = x.count(i)

Print(i, ":", n)

d1[i] = n

Print(d1)

WAP accept a statement and display frequency of words

x = input("Enter a sentence: ")

words = x.split()

Print(words)

d1 = {}

for i in set(words):

n = words.count(i)

Print(i, ":", n)

d1[i] = n

Print(d1)

WAP accept a file and display frequency of words in dictionary format

import os

fn = input("Enter a filename: ")

```

if os.path.exists(fn):
    if os.path.isfile(fn):
        with open(fn, "r") as x:
            k=x.read()
            words=k.split()
            fPrint(words)
            d1={}

            for i in set(words):
                n=words.count(i)
                #Print(i,":",n)
                d1[i]=n
            print(d1)

    else:
        print(fn, "is not a file.")

else:
    print(fn, "no such file or dir:")

```

If WAP accept a directory and count total no. of files and total no. of directories.

```

import os
dn=input("Enter directory name:")
if os.path.exists(dn):
    if os.path.isdir(dn):
        files=os.listdir(dn)
        print(files)
        fcnt=dcnt=0
        print(os.getcwd())
        for i in files:
            if os.path.isfile(i):
                fcnt+=1
            elif os.path.isdir(i):
                dcnt+=1
        print("Total no. of files:",fcnt)

```

Print ("Total no. of directories: ", &cnt)

else:

Point (dn, "is not a directory")

else:

Print (dn, "no such file or directory")

Varied List → List of List

1. $x = [[101, 'Ganesh', 'President', 300000], [102, 'Hari', 'Manager', 25000], [103, 'Lakshmi', 'Analyst', 28000]]$

Print (len(x)) # 3

Print (x[0]) # [102, 'Hari', 'Manager', 25000]

Print (x[0])

Print (x[2])

Print (x[1][2]) # 2nd emp sal, 28000

Print (x[0][2]) # 1st emp JOS, 'President'

Print (x[2][2]) # 2nd emp name, 'Lakshmi'

$x[1][2] = 30000$

$x[2].append(5000)$

$x.append([104, 'Sai', 'Analyst', 25000])$

Print x

WAP to find length without using len() function.

$x = [10, 20, 52, 3, 35]$

cnt = 0

for i in x:

cnt += 1

Print ("length:", cnt)

WAP Store all 1 digit nos into one list and 2 digit nos into another list and 3 digit numbers into another list. Other remaining numbers into another list. Display :

~~x = [10, 20, 503, 3, 35, 600, 73, 5, 200, 4568]~~

L₁ = []

L₂ = []

L₃ = []

L₄ = []

for i in x:

if i in range(0): L₁.append(i)

elif i in range(10, 100): L₂.append(i)

elif i in range(100, 1000): L₃.append(i)

else: L₄.append(i)

if i >= 0 and i <= 9: L₁.append(i)

elif i >= 10 and i <= 99: L₂.append(i)

elif i >= 100 and i <= 999: L₃.append(i)

else: L₄.append(i)

print("1 digit numbers list:", sorted(L₁))

print("2 digit numbers list:", sorted(L₂))

print("3 digit ...", sorted(L₃))

print("4 digit ...", sorted(L₄))

WAP Print elements which are ending
with 's' characters.

x = ['Unix', 'Ans', 'Linux', 'Perl', 'Python',
'Java', 'Pen', 'Linux', 'deng']

for i in x:

if i[-1] in "ss":

print(i)

or

y = [i for i in x if i[-1] in "ss"]

print(y)

WAP Print non duplicate elements

```
x = ['cent', 'aws', 'linux', 'Peri', 'python', 'Java',
      'peri', 'linux', 'devops']
```

```
for i in x:
    if x.count(i) == 1:
        print(i)
```

Q8

```
y = [i for i in x if x.count(i) == 1]
print(y)
```

WAP remove duplicate elements without using any function(set()) .

```
x = ['cent', 'aws', 'linux', 'Peri', 'Python', 'Java', 'peri',
      'linux', 'devops']
```

```
y = []
for i in x:
    if i not in y:
        y.append(i)
print(y)
```

WAP Print non duplicate elements

```
x = ['curl', 'curl', 'linux', 'Perl', 'Python', 'Java',
      'Perl', 'linux', 'devops']
for i in x:
    if x.count(i) == 1:
        print(i)
```

Q8

```
y = [i for i in x if x.count(i) == 1]
print(y)
```

WAP remove duplicate elements without using any function(set()) .

```
x = ['curl', 'curl', 'linux', 'Perl', 'Python', 'Java', 'Perl',
      'linux', 'devops']
y = []
for i in x:
    if i not in y:
        y.append(i)
print(y)
```

`t = (10,)` this is tuple.

enumerate → ~~Index~~ index + value

`()` → Tuple immutable item

`{ }` → set, does not support indexing
does not support ~~item~~ assignment

does not allow duplicates C/Hd of unique values

empty set `{} & set`

`isSet() → using set()`

`[] → list`

`frozenset([])`

immutable object

String, number, tuple, frozenset
mutable list, set, dictionary

Dictionary :-

group of key pair values

index → mean starting 0, 1, 2, ...

we can define our own index.

- like telephone dictionary
Name: phone no.

`x = { 'apithya' }`