In [6]:

```
lst = [3,5,3,2,2,6,6,6,3]
dic={}
for i in lst:
     dic[i]=lst.count(i)
print(dic)
```

```
{3: 3, 5: 1, 2: 2, 6: 3}
```

In [7]:

```
lst=[3,5,3,2,2,6,6,6,3]
dic={}
for i in lst:  #i=3,i=5,i=3,i=2,i=2
    if i not in dic:
        dic[i] = 1  #{3:1,5:1,2:1}
    else:
        dic[i]=dic[i]+1  #{3:2,5:1,2:1}
print(dic)
```

```
{3: 3, 5: 1, 2: 2, 6: 3}
```

In [15]:

```
lst=[3,5,3,2,2,6,6,6,3]
dic = {}
dic = dict()
for i in lst:
    dict.setdefault(i,lst.count(i))
print(dict)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-15-3e16aa25c30d> in <module>
      3 dic = dict()
      4 for i in lst:
----> 5     dict.setdefault(i,lst.count(i))
      6 print(dict)

TypeError: descriptor 'setdefault' requires a 'dict' object but received a
 'int'
```

Sets

- {}
- sets will remove the repitition

In [9]:

```
l = [1,2,4,3,5,2,1,3,4]
print(set(l))
```

Out[9]:

```
{1, 2, 3, 4, 5}
```

In [3]:

```
l = [1,2,4,3,5,2,1,3,4]
print(set(l))
list(set(l))
```

```
{1, 2, 3, 4, 5}
```

Out[3]:

```
[1, 2, 3, 4, 5]
```

In [4]:

```
s = {1,2,4,3,5,2,1,3,4}
type(s)
s
```

Out[4]:

```
{1, 2, 3, 4, 5}
```

In [5]:

```
s =l.copy()
```

In [6]:

```
l
```

Out[6]:

```
[1, 2, 4, 3, 5, 2, 1, 3, 4]
```

In [7]:

```
print(l)
l1 = []
for i in l:
    if i not in l1:
        l1.append(i)
print(l1)
```

```
[1, 2, 4, 3, 5, 2, 1, 3, 4]
[1, 2, 4, 3, 5]
```

In [8]:

```
print(l)
l1 = []
for i in l:
    if i not in l1:
        l1.append(i)
l1.sort()
print(l1)
```

```
[1, 2, 4, 3, 5, 2, 1, 3, 4]
[1, 2, 3, 4, 5]
```

In [9]:

```
print(dir(l))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__di
r__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__ge
titem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_
subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmu
l__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook
__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop',
'remove', 'reverse', 'sort']
```

In [13]:

```
l=[1,2,"a",3,5,"a",1,"b",4]
set(l)
```

Out[13]:

```
{1, 2, 3, 4, 5, 'a', 'b'}
```

In [11]:

```
print(dir(set))
```

```
['__and__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc_
_', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash_
_', '__iand__', '__init__', '__init_subclass__', '__ior__', '__isub__', '__i
ter__', '__ixor__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__o
r__', '__rand__', '__reduce__', '__reduce_ex__', '__repr__', '__ror__', '__r
sub__', '__rxor__', '__setattr__', '__sizeof__', '__str__', '__sub__', '__su
bclasshook__', '__xor__', 'add', 'clear', 'copy', 'difference', 'difference_
update', 'discard', 'intersection', 'intersection_update', 'isdisjoint', 'is
subset', 'issuperset', 'pop', 'remove', 'symmetric_difference', 'symmetric_d
ifference_update', 'union', 'update']
```

In [14]:

```
s = {1,2,3,4,5}
s.add(10)  # to add an element to the set
s
```

Out[14]:

```
{1, 2, 3, 4, 5, 10}
```

In [26]:

```
s1 = {1,2,3,4,5}
s2 = {10,20,20,1,2,3}
s1.add("apssdc")
s1
```

Out[26]:

```
{1, 2, 3, 4, 5, 'apssdc'}
```

In [27]:

```
copy_s=s.copy()
copy_s
```

Out[27]:

```
{1, 2, 3, 4, 5, 10}
```

In [28]:

```
copy_s
```

Out[28]:

```
{1, 2, 3, 4, 5, 10}
```

In [29]:

```
print(s1)
print(s2)
s1.difference(s2)
```

```
{1, 2, 3, 4, 5, 'apssdc'}
{1, 2, 3, 10, 20}
```

Out[29]:

```
{4, 5, 'apssdc'}
```

In [30]:

```
print(s1)
print(s2)
```

```
{1, 2, 3, 4, 5, 'apssdc'}
{1, 2, 3, 10, 20}
```

In [31]:

```
print(s1)
print(s2)
print()
print(s1.difference_update(s2))
print()
print(s1)
print(s2)
```

```
{1, 2, 3, 4, 5, 'apssdc'}
{1, 2, 3, 10, 20}

None

{4, 5, 'apssdc'}
{1, 2, 3, 10, 20}
```

In [36]:

```
s1 = {1,2,3,4,5,'Apssdc'}
s2 = {1,2,3,10,20}
s1=s1.intersection(s2)
```

In [37]:

```
print(s1)
```

```
{1, 2, 3}
```

In [38]:

```
s1.isdisjoint(s2)
```

Out[38]:

```
False
```

In [39]:

```
print(s1)
print(s2)
```

```
{1, 2, 3}
{1, 2, 3, 10, 20}
```

In [40]:

```
print(s1.issubset(s2))
print(s2.issuperset(s1))
```

```
True
True
```

In [41]:

```
s1 = {1,2,3,4,5,'Apssdc'}
s2 = {1,2,3,10,20}
s1.symmetric_difference(s2)
```

Out[41]:

```
{10, 20, 4, 5, 'Apssdc'}
```

In [42]:

```
s1.symmetric_difference_update(s2)
print(s1)
print(s2)
```

```
{4, 5, 10, 'Apssdc', 20}
{1, 2, 3, 10, 20}
```

In [43]:

```
s1.union(s2)
```

Out[43]:

```
{1, 10, 2, 20, 3, 4, 5, 'Apssdc'}
```

In [44]:

```
print(s1)
print(s2)
```

```
{4, 5, 10, 'Apssdc', 20}
{1, 2, 3, 10, 20}
```

In [45]:

```
s1.update(s2)  # to join two sets
s1
```

Out[45]:

```
{1, 10, 2, 20, 3, 4, 5, 'Apssdc'}
```

In [46]:

```
l = [1,2,4,3,5,2,1,3,4]
l.pop()
```

Out[46]:

```
4
```

In [47]:

```
l
```

Out[47]:

```
[1, 2, 4, 3, 5, 2, 1, 3]
```

In [48]:

```
s.pop()
```

Out[48]:

```
1
```

In [49]:

```
s
```

Out[49]:

```
{2, 3, 4, 5, 10}
```

In [ ]:

```

```

In [50]:

```
1 print(s.remove(4))
```

```
None
```

In [51]:

```
1 s
```

Out[51]:

```
{2, 3, 5, 10}
```

In [52]:

```
1 print(s.discard(3))
```

```
None
```

In [53]:

```
1 s
```

Out[53]:

```
{2, 5, 10}
```

In [56]:

```
1 print(s.discard(20))
2 print(s.remove(20))
```

```
None
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-56-3047e5db0486> in <module>
      1 print(s.discard(20))
----> 2 print(s.remove(20))

KeyError: 20
```

In [58]:

```
1 s.clear()
```

In [59]:

```
1 type(s)
```

Out[59]:

```
set
```

In [60]:

```
1 del s
```

In [61]:

```
1 s
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-61-ded5ba42480f> in <module>
----> 1 s

NameError: name 's' is not defined
```

In [64]:

```
1 s1 = {1,2,3,4,5,10,20}
2 s2 = {10,20,"a","b"}  #
3 s2.difference(s1)
4 s2.difference_update(s1)
5 s2
```

Out[64]:

```
{'a', 'b'}
```

In [65]:

```
1 s1.union(s2)
```

Out[65]:

```
{1, 10, 2, 20, 3, 4, 5, 'a', 'b'}
```

In [66]:

```
1 s1
```

Out[66]:

```
{1, 2, 3, 4, 5, 10, 20}
```

In [67]:

```
1 s1.update(s2)
2 s1
```

Out[67]:

```
{1, 10, 2, 20, 3, 4, 5, 'a', 'b'}
```

In [68]:

```
1 s1 = {1,2,3,4,5,10,20}
2 s2 = {10,20,"a","b"}  #{a,b}
3 s2.difference(s1)
4 s1
```

Out[68]:

```
{1, 2, 3, 4, 5, 10, 20}
```

In [69]:

```
s2.symmetric_difference(s1)
```

Out[69]:

```
{1, 2, 3, 4, 5, 'a', 'b'}
```

In [70]:

```
s2.symmetric_difference_update(s1)
```

In [71]:

```
s2
```

Out[71]:

```
{1, 2, 3, 4, 5, 'a', 'b'}
```

In [72]:

```
s = "Python programming"
s1 = ""
for i in s:
    if i not in s1:
        s1=s1+i
c=0
for j in s1:
    if j.isalpha():
        c=c+1
print(c)
```

```
12
```

In [ ]:

```

```