```
l = [5,[5,3,2,5],(5,3,7),10,(5,),20,5,[5]]

value = 5
count = 0
for i in l:
  if isinstance(i,int):
    if i == value:
      count = count + 1
  elif isinstance(i,list):
    for j in i:
      if j == value:
        count = count + 1
  elif isinstance(i,tuple):
    for j in i:
      if j == value:
        count = count + 1
else:
  print(value,'count :',count)
```

```
5 count : 7
```

set

set is mutable

set is a sequence

set is iterable

set can store any number of values or items

set can store only immutable data type values or items

set can not store duplicate values or items

set does not support indexing

set does not support slicing

set is unordered

set items enclosed within { , , , }

empty set : set( )

```
s = {1,2,3,4,5,6}
print(s)
```

```
{1, 2, 3, 4, 5, 6}
```

```
print(type(s))
```

```
<class 'set'>
```

```
print(id(s))
```

```
139954043052992
```

```
print(len(s))
```

```
6
```

```
for i in s:
  print(i,end=' ')
```

```
1 2 3 4 5 6
```

```
print(s[0])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-9-5605e14cbe49> in <module>
----> 1 print(s[0])

TypeError: 'set' object is not subscriptable

SEARCH STACK OVERFLOW
```

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

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

```
s = set()
print(s)
```

```
set()
```

```
s = {10,True,12.89,(1,2,3,4),'python',[6,2,8],{1,2}}
print(s)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-12-406216296c7b> in <module>
----> 1 s = {10,True,12.89,(1,2,3,4),'python',[6,2,8],{1,2}}
      2 print(s)

TypeError: unhashable type: 'list'

SEARCH STACK OVERFLOW
```

```
s = {10,True,12.89,(1,2,3,4),'python',{1,2}}
print(s)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-13-c4d4376f509a> in <module>
----> 1 s = {10,True,12.89,(1,2,3,4),'python',{1,2}}
      2 print(s)

TypeError: unhashable type: 'set'

SEARCH STACK OVERFLOW
```

```
s = {10,True,12.89,(1,2,3,4),'python'}
print(s)
```

```
{True, 10, 12.89, 'python', (1, 2, 3, 4)}
```

```
s = {'srinath',True,10,'srikanth',12.45,1,2,3}
print(s)
```

```
{True, 2, 3, 10, 12.45, 'srinath', 'srikanth'}
```

```
s = {'srinath',1,10,'srikanth',12.45,True,2,3}
print(s)
```

```
{1, 2, 3, 10, 12.45, 'srinath', 'srikanth'}
```

```
s = {'srinath',10,'srikanth',12.45,True,2,1,3}
print(s)
```

```
{True, 2, 3, 10, 12.45, 'srinath', 'srikanth'}
```

set.add( )

```
s = {1,5,2,7,9}
s.add(2)
print(s)
```

```
{1, 2, 5, 7, 9}
```

```
s = {1,5,2,7,9}
s.add(20)
print(s)
```

```
{1, 2, 5, 7, 9, 20}
```

```
s = {1,5,2,7,9}
s.add(-10)
print(s)
```

```
{1, 2, 5, 7, 9, -10}
```

```
s = {1,5,2,7,9}
s.add(False)
print(s)
```

```
{False, 1, 2, 5, 7, 9}
```

```
s = {1,5,2,7,9}
s.add(0)
print(s)
```

```
{0, 1, 2, 5, 7, 9}
```

```
s = {1,5,False,2,7,9}
s.add(0)
print(s)
```

```
{False, 1, 2, 5, 7, 9}
```

```
s = {1,5,2,7,9}
s.add((4,2,7))
print(s)
```

```
{1, 2, 5, 7, 9, (4, 2, 7)}
```

```
s = {1,5,2,7,9}
s = list(s)
print(s)
print(type(s))
```

```
[1, 2, 5, 7, 9]
<class 'list'>
```

```
s = {1,5,2,7,9}
s = tuple(s)
print(s)
print(type(s))
```

```
(1, 2, 5, 7, 9)
<class 'tuple'>
```

```
s = {2,5,7,(10,12,20),True}
for i in s:
  if isinstance(i,tuple):
    for j in i:
      print(j,end=' ')
```

```
10 12 20
```

set.update( )

```
s = {1,2,3,4,5}
s.update(10,20)
print(s)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-28-e1a48451828a> in <module>
      1 s = {1,2,3,4,5}
----> 2 s.update(10,20)
      3 print(s)

TypeError: 'int' object is not iterable
```

SEARCH STACK OVERFLOW

```
s = {1,2,3,4,5}
s.update([10,20])
print(s)
```

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

```
s = {1,2,3,4,5}
s.update((10,20))
print(s)
```

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

```
s = {1,2,3,4,5}
s.update('python')
print(s)
```

```
{'h', 1, 2, 3, 4, 5, 't', 'n', 'p', 'y', 'o'}
```

```
print(s)
```

```
{'h', 1, 2, 3, 4, 5, 't', 'n', 'p', 'y', 'o'}
```

```
s = {1,2,3,4,5}
s1 = {5,6,7,8,9}
s.update(s1)
print(s)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

set.remove( )

```
s = {1,2,3,4,5,6,7,8,9,10}
s.remove(5)
print(s)
```

```
{1, 2, 3, 4, 6, 7, 8, 9, 10}
```

```
s = {1,2,3,4,5,6,7,8,9,10}
s.remove(15)
```

```
print(s)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-36-992abd6d3118> in <module>
      1 s = {1,2,3,4,5,6,7,8,9,10}
----> 2 s.remove(15)
      3 print(s)

KeyError: 15
```

SEARCH STACK OVERFLOW

set.discard( )

```
s = {1,2,3,4,5,6,7,8,9,10}
s.discard(3)
```

```
s = {1,2,3,4,5,6,7,8,9,10}
s.discard(500)
```

set.pop( )

```
s = {1,2,3,4,5,6,7,8,9,10}
print(s.pop())
print(s)
```

```
1
{2, 3, 4, 5, 6, 7, 8, 9, 10}
```

set.clear( )

```
s = {1,2,3,4,5,6,7,8,9,10}
s.clear()
print(s)
```

```
set()
```

set.union( )

```
A = {1,2,3}
B = {3,4,5}
print(A.union(B))
```

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

```
A = {1,2,3}
B = {3,4,5}
print(A|B)
```

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

set.intersection( )

```
A = {1,2,3}
B = {3,4,5}
print(A&B)
```

```
{3}
```

```
A = {1,2,3}
B = {3,4,5}
print(A.intersection(B))
```

```
{3}
```

set.difference( )

```
A = {1,2,3}
B = {3,4,5}
print(A.difference(B))
print(A-B)
```

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

```
A = {1,2,3}
B = {3,4,5}
print(B.difference(A))
print(B-A)
```

```
{4, 5}
{4, 5}
```

set.symmetric_difference( )

```
A = {1,2,3}
B = {3,4,5}
print(A.symmetric_difference(B))
```

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

set.intersection_update( )

```
A = {1,2,3}
B = {3,4,5}
A.intersection_update(B)
print(A)
print(B)
```

```
{3}
{3, 4, 5}
```

set.difference_update( )

```
A = {1,2,3}
B = {3,4,5}
A.difference_update(B)
print(A)
print(B)
```

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

set.symmetric_difference_update( )

```
A = {1,2,3}
B = {3,4,5}
A.symmetric_difference_update(B)
```

```
print(A)
print(B)
```

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

set.isdisjoint( )

```
A = {1,2,3}
B = {3,4,5}
print(A.isdisjoint(B))
```

```
False
```

set.issuperset( )

```
A = {1,2,3}
B = {3,4,5}
print(A.issuperset(B))
```

```
False
```

```
A = {1,2,3}
B = {3}
print(A.issuperset(B))
```

```
True
```

set.issubset( )

```
A = {1,2,3}
B = {3,4,5}
print(A.issubset(B))
```

```
False
```

```
A = {1,2,3}
B = {3,4,5,1,2}
print(A.issubset(B))
```

```
True
```

```
l = [1,4,2,3,2,4,6,8,6,5,4,7,6,3,2,3]
# create a new list with unique values
# without using type-conversion

l = [1,4,2,3,2,4,6,8,6,5,4,7,6,3,2,3]
# [1,4,6,3,8,12,18,26,32,5,36,7,42,3,44,3]
```

✓ 0s completed at 2:02 PM