In [1]:
```
#Filtering the Elements of ndarray
```

In [2]:
```
import numpy as np
```

In [3]:
```
lst=[10,-20,34,-45,67,-38,-12,0,13]
a=np.array(lst)
print(a,type(a))
```

```
[ 10 -20  34 -45  67 -38 -12   0  13] <class 'numpy.ndarray'>
```

In [4]:
```
#Step-1: Prepare boolean array with Condition----ndarray with condition
ba=a>0
print(ba,type(ba))
```

```
[ True False  True False  True False False False  True] <class 'numpy.ndarra
y'>
```

In [5]:
```
#Step-2: Pass the boolean array to ndarray object-----ndarrayobj[boolean array
a[ba]
```

Out[5]:
```
array([10, 34, 67, 13])
```

In [6]:
```
#OR-Direct Approach
#Syntax: ndarrayobj[ndarrayobject with condition]
a[a>0]
```

Out[6]:
```
array([10, 34, 67, 13])
```

In [7]:
```
a[a<0]
```

Out[7]:
```
array([-20, -45, -38, -12])
```

In [8]:
```
lst=[10,20,30,40,50,60,70,80,90]
a=np.array(lst)
a.shape=(3,3)
print(a)
```

```
[[10 20 30]
 [40 50 60]
 [70 80 90]]
```

In [10]:
```
#obtain multiples of 3
a[a%3==0]
```

Out[10]:
```
array([30, 60, 90])
```

```
In [12]: #obtain multiples of 4
         a[a%4==0]

Out[12]: array([20, 40, 60, 80])
```

```
In [13]: #obtain multiples of 5
         a[a%5==0]

Out[13]: array([10, 20, 30, 40, 50, 60, 70, 80, 90])
```

```
In [14]: a[(a>20)&(a<55)] # Use Bitwise Operators but not Logical Operators

Out[14]: array([30, 40, 50])
```

```
In [15]: # a[20<a<55] # Invalid

         ---------------------------------------------------------------------------
         ValueError                                Traceback (most recent call last)
         Cell In[15], line 1
         ----> 1 a[20<a<55]

         ValueError: The truth value of an array with more than one element is ambiguo
         us. Use a.any() or a.all()
```

```
In [ ]:
```