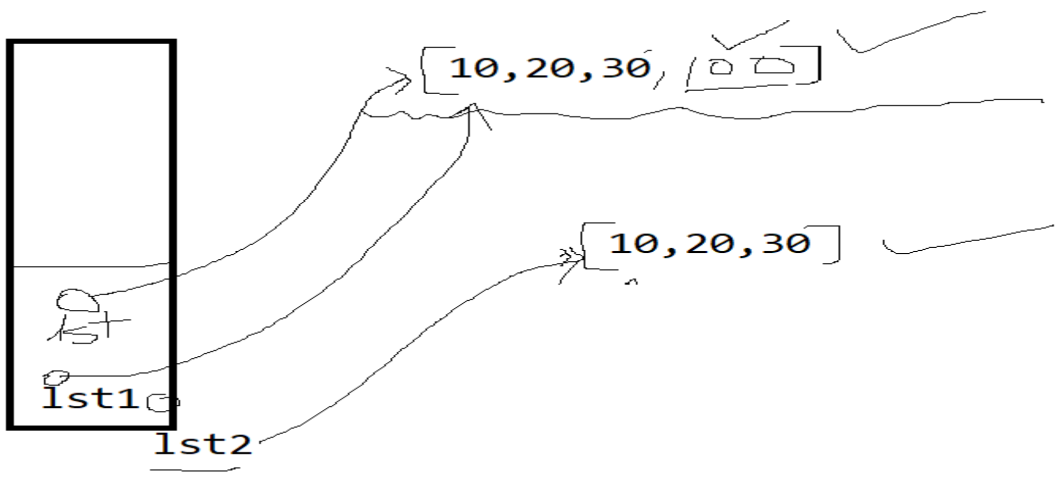
List

1. It is a heterogeneous collection of data
2. It is ordered collection.
3. It allows to access data by index, and hence random access is possible
4. It allows to store duplicate values.
5. It is mutable.
6. It is represented using []

Lst.append(value)	It adds single value at the end of the list
Lst.extend(iterable)	It adds all values at the end from iterable one by one in the lst
Lst.insert(pos,value)	It will add the value at the given position, if position is out of bounds, then it will insert at the end
Lst.pop([pos])	It will delete the data from last index position if the pos is not given, otherwise it deletes from the given position
Lst.remove(value)	It will delete the first occurrence of the given value from the list if found, otherwise it throws an exception
Del(lst[pos])	It deletes the data from the given position
Lst.index(value,[start,end])	It will give you the position of the first occurrence if found, otherwise it throws an exception
Lst.clear()	It removes all the data from the list, and keeps the list with 0 length
Lst.reverse()	It will reverse order of the list, but it changes the original list
Lst.sort()	It will sort the list only if the list is homogenous, it changes the original list
Lst.copy()	It is used to create a shallow copy of the list

Copy function in the list creates a shallow copy



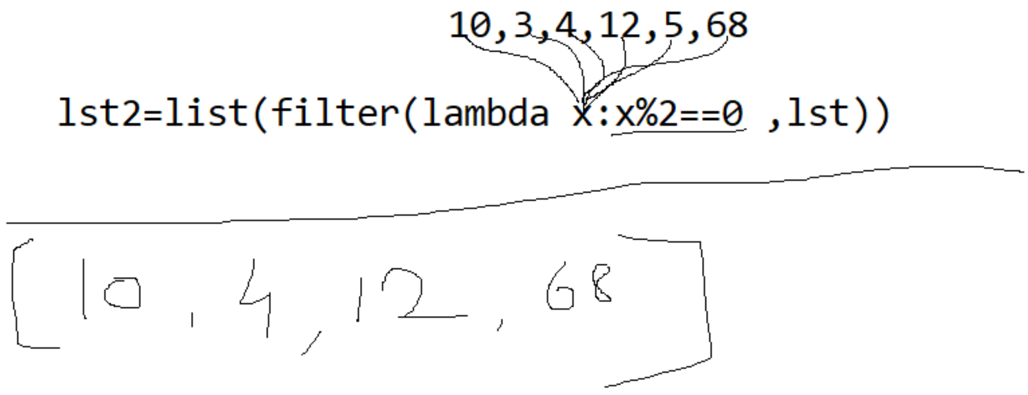
lst=[12,23,34]

lst1=lst

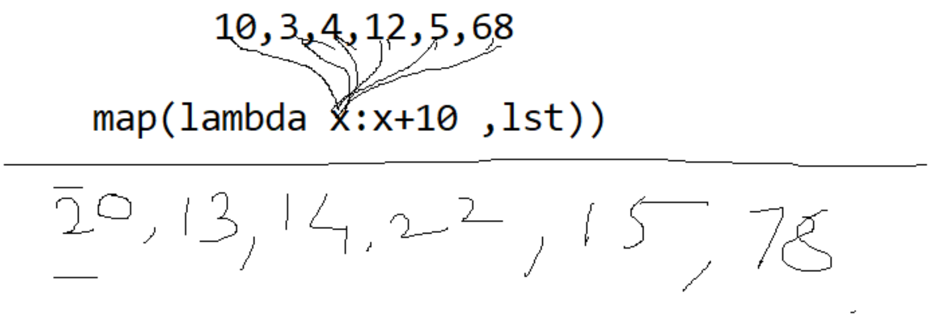
lst.append(100)

```
print(lst)

print(lst1)


lst2=lst.copy()

lst.append(200)

print(lst,lst1)

print(lst2)
```

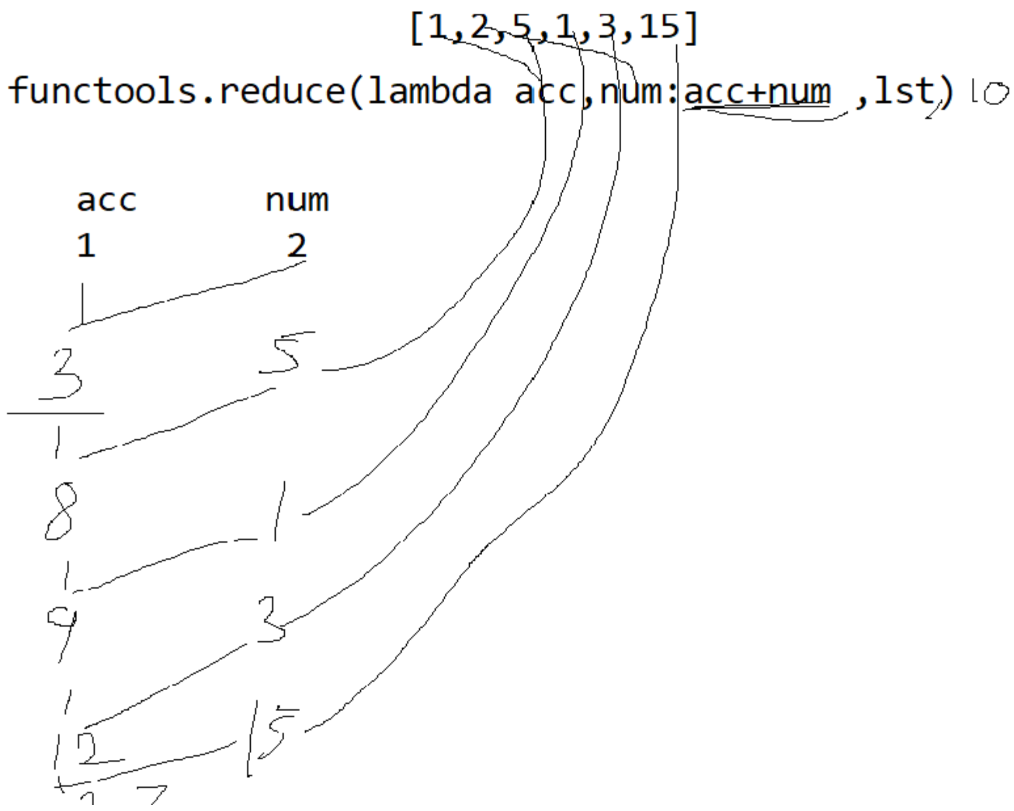
Zip(lst,lst1,lst2,lst3)	It is used to read data from multiple lists simulteneouly



```
lst2=list(filter(lambda x:x%2==0 ,lst))
```



```
map(lambda x:x+10 ,lst))
```



```
[1,2,5,1,3,15]
functools.reduce(lambda acc,num:acc+num ,lst)
```