extend()
this method allows to add more than one value to list
Syntax:
<list-name>.extend(sequence/iterable)

```
>>> list1=list(range(10,60,10))
>>> print(list1)
[10, 20, 30, 40, 50]
>>> list1.extend([60,70,80,90,100])
>>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

pop()

pop() method return and remove last element of list.
if there is no more elements to pop it raises error.

Syntax: pop()
Syntax: pop(index)

List can be used as stack. Stack is a data structure which follows LIFO (Last In First Out), the element added last is removed first.

Stack allows 2 operations

1. Push
2. Pop

Push → adding element/item (append)
Pop → removing element/item (pop)

Example:

```
# implementation of stack using list

stack=[]
while True:
    print("1.push")
    print("2.pop")
    print("3.view")
    print("4.exit")
    opt=int(input("enter any option"))
```

```
    if opt==1:
        ele=int(input("enter any element"))
        stack.append(ele)
        print("element pushed inside stack")
    elif opt==2:
        if len(stack)==0:
            print("stack is empty")
        else:
            ele=stack.pop()
            print("element poped is ",ele)
    elif opt==3:
        print(stack)
    elif opt==4:
        break
```

Output:

```
1.push
2.pop
3.view
4.exit
enter any option1
enter any element10
element pushed inside stack
1.push
2.pop
3.view
4.exit
enter any option1
enter any element20
element pushed inside stack
1.push
2.pop
3.view
4.exit
enter any option1
```

Example:

```
# implementation of queue using list

queue=[]
```

```
while True:
    print("1.add")
    print("2.remove")
    print("3.view")
    print("4.exit")
    opt=int(input("enter your option"))
    if opt==1:
        ele=int(input("enter any element"))
        queue.append(ele)
        print("element added")
    elif opt==2:
        if len(queue)==0:
            print("queue is empty")
        else:
            ele=queue[0]
            del queue[0]
            print(ele,"removed from queue")
    elif opt==3:
        print(queue)
    elif opt==4:
        break
```

Output:

```
1.add
2.remove
3.view
4.exit
enter your option1
enter any element10
element added
1.add
2.remove
3.view
4.exit
enter your option1
enter any element20
element added
1.add
2.remove
```

3.view
4.exit
enter your option1
enter any element30
element added

clear()
this method is used to empty list or to remove all elements from list

```
>>> list1=[10,20,30,40,50]
>>> print(list1)
[10, 20, 30, 40, 50]
>>> list1.clear()
>>> print(list1)
[]
>>> list2=[10,20,30,40,50]
>>> print(list2)
[10, 20, 30, 40, 50]
>>> del list2[:]
>>> print(list2)
[]
```

sort()
sort(*, key=None, reverse=False)
This method sorts the list in place, using only < comparisons between items.

```
>>> list1=[6,1,5,2,9,8,4,7]
>>> list1.sort()
>>> print(list1)
[1, 2, 4, 5, 6, 7, 8, 9]
>>> list1.sort(reverse=True)
>>> print(list1)
[9, 8, 7, 6, 5, 4, 2, 1]
```

```
>>> list2=['a','B','A','c','b','C']
>>> list2.sort()
>>> print(list2)
['A', 'B', 'C', 'a', 'b', 'c']
>>> list2.sort(reverse=True)
>>> print(list2)
['c', 'b', 'a', 'C', 'B', 'A']
>>> list2.sort(key=str.upper)
>>> print(list2)
['a', 'A', 'b', 'B', 'c', 'C']
```

reverse()
This method reverse the elements of list

```
>>> list1=[10,20,30,40,50]
>>> print(list1)
[10, 20, 30, 40, 50]
>>> list1.reverse()
>>> print(list1)
[50, 40, 30, 20, 10]
>>> list2=list1[::-1]
>>> print(list2)
```

https://www.hackerrank.com/challenges/python-lists/problem?isFullScreen=false