

Example of all occurrences of 10

```
list1=[10,20,30,10,10,40,50,10,20,10,10]
while True:
    if 10 in list1:
        list1.remove(10)
    else:
        break

print(list1)
```

Output

```
[20, 30, 40, 50, 20]
```

clear() : This method remove all the values from list or empty list

```
>>> list1=list(range(10,110,10))
>>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list1.clear()
>>> print(list1)
[]
>>> list2=list(range(10,110,10))
>>> print(list2)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> del list2[:]
>>> print(list2)
[]
```

pop() : This method remove last value from list. It uses LIFO (Last In First Out). Pop method before removing, returns value and remove it. List can be used as STACK. Stack is a data structure which follows LIFO (Last In First Order).

Example:

```
>>> list1=[10,20,30,40,50]
```

```
>>> list1.pop()
50
>>> print(list1)
[10, 20, 30, 40]
>>> list1.pop()
40
>>> print(list1)
[10, 20, 30]
>>> x=list1.pop()
>>> print(x)
30
>>> print(list1)
[10, 20]
```

Example

```
# Implementation of Stack using list
# Stack allows two operations
# 1. push --> Adding element
# 2. pop --> Removing element
```

```
stack=[]
while True:
    print("1. Push")
    print("2. Pop")
    print("3. Display")
    print("4. Exit")
    opt=int(input("Enter your option :"))
    if opt==1:
        value=int(input("Enter Value "))
        stack.append(value)
        print(f'{value} pushed inside stack')
    elif opt==2:
        if len(stack)==0:
            print("Stack is empty")
        else:
            value=stack.pop()
            print(f'{value} popped from stack')
```

```
elif opt==3:
    print(f'Stack is {stack}')
elif opt==4:
    print("bye...")
    break
```

Output

```
1. Push
2. Pop
3. Display
4. Exit
Enter your option :1
Enter Value 10
10 pushed inside stack
```

Example

```
# Implementing Queue using list
# Queue follows FIFO (First In First Out), The value inserted first is
removed first
# Queue allows 2 operations
# 1. adding
# 2. removing

queue=[]
while True:
    print("1. Add")
    print("2. Remove")
    print("3. Display")
    print("4. Exit")
    opt=int(input("Enter Your Option :"))
    match(opt):
        case 1:
            value=int(input("Enter value "))
            queue.append(value)
            print(f'{value} is added to queue')
        case 2:
            if len(queue)==0:
```

```
        print("Queue is empty")
    else:
        value=queue[0]
        del queue[0]
        print(f'{value} removed from queue')
    case 3:
        print(f'Queue values {queue}')
    case 4:
        break
```

Output

```
1. Add
2. Remove
3. Display
4. Exit
Enter Your Option :1
Enter value 10
10 is added to queue
```

Inserting value within list

Inserting value within list are done 2 ways

1. Using insert method
2. Using slicing operator

Insert method

This method inserts one value.

This method inserts value at given index.

Syntax: list-name.insert(index,value)

Example:

```
>>> list1=[10,20,30,40,50]
>>> print(list1)
[10, 20, 30, 40, 50]
>>> list1.insert(0,99)
>>> print(list1)
[99, 10, 20, 30, 40, 50]
```

```
>>> list1.insert(2,88)
>>> print(list1)
[99, 10, 88, 20, 30, 40, 50]
>>> list1.insert(-2,77)
>>> print(list1)
[99, 10, 88, 20, 30, 77, 40, 50]
>>> list1.insert(10,66)
>>> print(list1)
[99, 10, 88, 20, 30, 77, 40, 50, 66]
>>> list1.insert(-20,44)
>>> print(list1)
[44, 99, 10, 88, 20, 30, 77, 40, 50, 66]
```

Inserting multiple values using slicing operation

Syntax: list-name[start-index:stop-index]=iterable

Note: start and stop index must be same

```
>>> list1=[10,20,30,40,50]
>>> list1[0:0]=[88,99]
>>> print(list1)
[88, 99, 10, 20, 30, 40, 50]
>>> list1[2:2]=[55,66,77]
>>> print(list1)
[88, 99, 55, 66, 77, 10, 20, 30, 40, 50]
>>> list1[-2:-2]=range(1,4)
>>> print(list1)
[88, 99, 55, 66, 77, 10, 20, 30, 1, 2, 3, 40, 50]
>>> list1[0:2]=[11,22]
>>> print(list1)
[11, 22, 55, 66, 77, 10, 20, 30, 1, 2, 3, 40, 50]
>>>
```

<https://www.geeksforgeeks.org/python-programming-examples/#list>

Copy or Cloning