**sort()  method**

It is a predefined method of list, this method sort the elements/values of list in ascending or descending order in place or in same list
It is a mutable method of list.

**Syntax:** list-name.sort(key=None,reverse=False)

By default sort method of list arrange the elements/values is ascending order, because default value of reverse=False
In order to arrange elements/values in descending order define value of reverse=True

Key is nothing but function which is applied to each element of list for comparing values

**Example:**

```
>>> A=[20,80,40,60,20,70,50,30]
>>> print(A)
[20, 80, 40, 60, 20, 70, 50, 30]
>>> A.sort()
>>> print(A)
[20, 20, 30, 40, 50, 60, 70, 80]
>>> A.sort(reverse=True)
>>> print(A)
[80, 70, 60, 50, 40, 30, 20, 20]
>>> B=["D","a","C","A","b","B","c"]
>>> print(B)
['D', 'a', 'C', 'A', 'b', 'B', 'c']
>>> B.sort()
>>> print(B)
['A', 'B', 'C', 'D', 'a', 'b', 'c']
>>> B.sort(reverse=True)
>>> print(B)
['c', 'b', 'a', 'D', 'C', 'B', 'A']
>>> B.sort(key=str.upper)
>>> print(B)
['a', 'A', 'b', 'B', 'c', 'C', 'D']
>>> B.sort(key=str.lower)
>>> print(B)
```

```
['a', 'A', 'b', 'B', 'c', 'C', 'D']
>>> B.sort(key=str.lower,reverse=True)
>>> print(B)
['D', 'c', 'C', 'b', 'B', 'a', 'A']
>>> C=["abcd","a","abcde","abc","ab"]
>>> print(C)
['abcd', 'a', 'abcde', 'abc', 'ab']
>>> C.sort(key=len)
>>> print(C)
['a', 'ab', 'abc', 'abcd', 'abcde']
>>> C.sort(key=len,reverse=False)
>>> C.sort(key=len,reverse=True)
>>> print(C)
['abcde', 'abcd', 'abc', 'ab', 'a']
```

**Example**

```
# Write a program in given list find first maximum,minimum

rohit=[20,10,0,5,50,50,70,30]
print(f'Before Sorting {rohit}')
rohit.sort()
print(f'After Sorting {rohit}')
print(f'Minimum score is {rohit[0]}')
print(f'Maximum score is {rohit[-1]}')
```

**Output**

```
Before Sorting [20, 10, 0, 5, 50, 50, 70, 30]
After Sorting [0, 5, 10, 20, 30, 50, 50, 70]
Minimum score is 0
Maximum score is 70
```

**Example:**

```
# Write a program in a given list scores find
# maximum score (1st maximum), runner up score (2nd maximum)

players=[40,10,50,60,30,70,20,70,60]
print(f'Players Score {players}')
players.sort()
print(f'After Sorting Players Score {players}')
print(f'First Maximum {players[-1]}')
```

```
first_max=players[-1]
c=0
for value in players:
    if value==first_max:
        c=c+1
print(f'Second Maximum is {players[-(c+1)]}')
```

**Output**

```
Players Score [40, 10, 50, 60, 30, 70, 20, 70, 60]
After Sorting Players Score [10, 20, 30, 40, 50, 60, 60, 70, 70]
First Maximum 70
Second Maximum is 60
```

**Example:**

```
# Write a program to input n elements or values in list
# and find median

A=[]
n=int(input("How many integer values?"))
for i in range(n):
    value=int(input("Enter any value "))
    A.append(value)

A.sort()
if n%2!=0:
    i=n//2
    median=A[i]
else:
    i=n//2
    x=A[i]
    y=A[i-1]
    median=(x+y)/2

print(f'median is {median:.2f}')
```

**Output**

```
How many integer values?6
Enter any value 1
Enter any value 2
Enter any value 3
```

```
Enter any value 4
Enter any value 5
Enter any value 6
median is 3.50

How many integer values?5
Enter any value 1
Enter any value 2
Enter any value 3
Enter any value 4
Enter any value 5
median is 3.00
```

**sorted() function**

This is immutable function
This function after sorted elements in returns sorted elements into new collection or iterable.
In application development sorted function is used,

1. With immutable collection, which does not provides sort() method
2. It is also used with mutable collection to return sorted elements into another collection

**Syntax:** <variable-name>=sorted(iterable,key=None,reverse=False)

```
>>> A=[5,3,1,4,2]
>>> print(A)
[5, 3, 1, 4, 2]
>>> B=sorted(A)
>>> print(B)
[1, 2, 3, 4, 5]
>>> print(A)
[5, 3, 1, 4, 2]
>>> C=sorted(A,reverse=True)
>>> print(C)
[5, 4, 3, 2, 1]
>>> print(A)
[5, 3, 1, 4, 2]
>>> D=["45","12","55","35","27","12"]
>>> print(D)
```

```
['45', '12', '55', '35', '27', '12']
>>> E=sorted(D,key=int)
>>> print(E)
['12', '12', '27', '35', '45', '55']
>>> F=sorted(D,key=int,reverse=True)
>>> print(F)
['55', '45', '35', '27', '12', '12']
```

**Immutable Operations of sequences**

**sequence-name.count(value)**

This method returns count of value in given sequence.
Returns number of occurrences of given value

```
>>> A=[1,2,3,1,1,1,1,2,4,5,5]
>>> print(A)
[1, 2, 3, 1, 1, 1, 1, 2, 4, 5, 5]
>>> A.count(1)
5
>>> A.count(2)
2
>>> A.count(3)
1
>>> A.count(4)
1
>>> A.count(5)
2
```

**Example**

```
# Write a program to find count of each value in list
# input ==> [1,2,3,2,3,2,2,4,5,5,5,5]
# output ==>1 -->1
#         2 -->3
#         3 -->2
#         4 -->1
#         5 -->4
A=[1,2,3,2,3,2,2,4,5,5,5,5]
B=[]
```

```
print(A)
for value in A:
    if value not in B:
        B.append(value)

print(B)
for value in B:
    c=A.count(value)
    print(f'{value}-->{c}')
```

**Output**

```
[1, 2, 3, 2, 3, 2, 2, 4, 5, 5, 5, 5]
[1, 2, 3, 4, 5]
1-->1
2-->4
3-->2
4-->1
5-->4
```

**sequence-name.index(value,start=0,stop=len(sequence))**

This method return index of first occurrence of value