Example:

```
# write a program to read n elements inside list
# and separate even numbers and odd number

n=int(input("enter the value of n")) # 5
list1=[]
for i in range(n):
    value=int(input("enter any value"))
    list1.append(value)

print(list1)
evenlist=[]
oddlist=[]
for value in list1:
    if value%2==0:
        evenlist.append(value)
    else:
        oddlist.append(value)


print(evenlist)
print(oddlist)
```

Output:

```
enter the value of n5
enter any value1
enter any value2
enter any value3
enter any value4
enter any value5
[1, 2, 3, 4, 5]
[2, 4]
[1, 3, 5]
```

slicing

Using index we can read one value
Using slicing we can read more than one value
Slicing allows read a list from list (sublist)
Slicing is done in two ways.

1. Using slice operator
2. Using slice object

Slicing required 3 values

1. Start index
2. Stop index
3. Step

Slicing internally uses range for generating indexes

Using slicing operator
Syntax:
list-name[start:stop:step]

Syntax-1: list-name[::]
Syntax-2: list-name[::step]
Syntax-3: list-name[start::]
Syntax-4: list-name[:stop:]
Syntax-5: list-name[start:stop]
Syntax-6: list-name[start:stop:step]
Syntax-7: list-name[:stop:step]
Syntax-8: list-name[start::step]

Syntax-1: list-name[::]
Default step value is +ve → +1
Default start value is → 0
Default stop value is → length of sequence/list

Syntax always read the values from left right

This syntax will create copy of the list

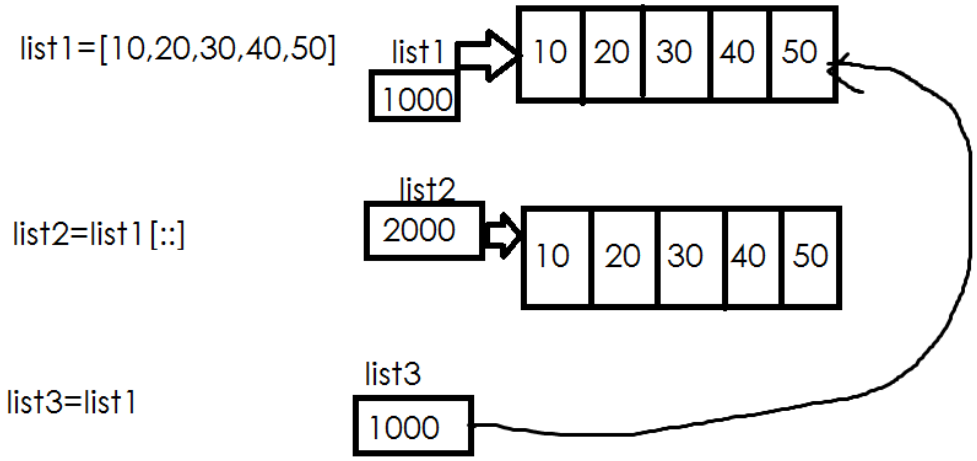
Example:

```
>>> list1=[10,20,30,40,50,60,70,80,90,100]
>>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list2=list1[::]
>>> print(list2)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list3=list1[:]
>>> print(list3)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

Syntax-2: list-name[::step]

In this syntax start, stop values are taken based step value
If step is +ve, the start=0,stop=length of list
If step is –ve, the start=-1,stop=-(length of list+1)

```
>>> list1=list(range(100,111))
>>> print(list1)
[100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110]
>>> list2=list1[::1]
>>> print(list2)
[100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110]
>>> list3=list1[::2]
>>> print(list3)
[100, 102, 104, 106, 108, 110]
>>> list4=list1[::12]
>>> print(list4)
[100]
>>> list5=list1[::-1]
>>> print(list5)
[110, 109, 108, 107, 106, 105, 104, 103, 102, 101, 100]
>>> list6=list1[::-2]
>>> print(list6)
[110, 108, 106, 104, 102, 100]
```



Syntax-3: list-name[start::]

In this syntax, default step is +1 and stop is till end of list

```
>>> l1=list(range(10,110,10))
>>> print(l1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> l2=l1[0::]
>>> print(l2)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> l3=l1[3::]
>>> print(l3)
[40, 50, 60, 70, 80, 90, 100]
>>> l4=l1[-3::]
>>> print(l4)
[80, 90, 100]
>>> l5=l1[-5:]
>>> print(l5)
[60, 70, 80, 90, 100]
```

Syntax-4: list-name[:stop:]

In this syntax default start=0 and step=+1

If stop is –ve, it converts to +ve (lengthof list-stop)

```
>>> l1=list(range(10,110,10))
>>> print(l1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> l2=l1[:5:]
```

```
>>> print(l2)
[10, 20, 30, 40, 50]
>>> l3=l1[:-5:]
>>> print(l3)
```

Syntax-5: list-name[start:stop]

```
>>> list1=list(range(10,110,10))
>>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list2=list1[0:5]
>>> print(list2)
[10, 20, 30, 40, 50]
>>> list3=list1[5:10]
>>> print(list3)
[60, 70, 80, 90, 100]
>>> list4=list1[3:-3]
>>> print(list4)
[40, 50, 60, 70]
>>> list5=list1[1:-1]
>>> print(list5)
[20, 30, 40, 50, 60, 70, 80, 90]
>>>
```