Slice operator
Syntax1: list-name[start:stop:step]
Syntax2: list-name[start::]
Syntax-3: list-name[:stop:]
Syntax4: list-name[::step]
Syntax5:list-name[start:stop]
Syntax6: list-name[start::step]
Syntax7: list-name[:stop:step]
Syntax8: list-name[::]
Syntax9: list-name[:]

Syntax8: list-name[::]
This syntax uses default values
start: 0
stop: length of list
step: +1
This syntax read all the value from list left to right (OR) create copy of the list

```
>>> list1=list(range(10,110,10))
>>> 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]
>>>
```

Syntax9: list-name[:]
It is same as syntax-8

```
>>> list1=list(range(10,210,20))
>>> print(list1)
[10, 30, 50, 70, 90, 110, 130, 150, 170, 190]
>>> list2=list1[:]
>>> print(list2)
[10, 30, 50, 70, 90, 110, 130, 150, 170, 190]
>>>
```

Syntax4: list-name[::step]
In this syntax start and stop values are taken based on step
If step +ve start=0,stop=len(list)

If step –ve start=-1,stop=-(len(list)+1)

Example:

```
>>> courses_list=['python','java','c','c++','oracle','mysql']
>>> courses_list1=courses_list[::1]
>>> courses_list2=courses_list[::2]
>>> print(courses_list1)
['python', 'java', 'c', 'c++', 'oracle', 'mysql']
>>> print(courses_list2)
['python', 'c', 'oracle']
>>>
>>> list1=list(range(10,110,10))
>>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list2=list1[::-1]
>>> print(list2)
[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]
>>> list3=list1[::-2]
>>> print(list3)
[100, 80, 60, 40, 20]
>>>
```

Syntax2: list-name[start::]
Default values of step:+1 and stop: len(list)
This syntax read elements from left to right

```
>>> list1=list(range(10,160,10))
>>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150]
>>> list2=list1[4:]
>>> print(list2)
[50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150]
>>> list3=list1[-4:]
>>> print(list3)
[120, 130, 140, 150]
>>>
```

Syntax-3: list-name[:stop:]
Stop value can be +ve or –ve
If stop is –ve, it converts into +ve stop

Note: len(list)-stopindex

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

Syntax5: 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[3:-2]
>>> print(list3)
[40, 50, 60, 70, 80]
>>> list4=list1[-2:-5]
>>> print(list4)
[]
>>>
```

https://csiplearninghub.com/important-practice-questions-of-list-in-python/

Syntax6: list-name[start::step]
Syntax7: list-name[:stop:step]

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

```
>>> list2=list1[0::1]
>>> print(list2)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list3=list1[0::2]
>>> print(list3)
[10, 30, 50, 70, 90]
>>> list4=list1[-2::1]
>>> print(list4)
[90, 100]
>>> list5=list1[-2::-1]
>>> print(list5)
[90, 80, 70, 60, 50, 40, 30, 20, 10]
>>>
```

Syntax1: list-name[start:stop:step]

```
>>> list1=list("PROGRAMMING")
>>> print(list1)
['P', 'R', 'O', 'G', 'R', 'A', 'M', 'M', 'I', 'N', 'G']
>>> list2=list1[3:6:1]
>>> print(list2)
['G', 'R', 'A']
>>> list3=list1[-3:-6:-1]
>>> print(list3)
['I', 'M', 'M']
>>>
```

Slice object

Slice operator is used only one time on list
Slice object persist/save/store start index, stop index and step
Syntax1: slice(stop)
Syntax2: slice(start,stop,[step])

```
S1=slice(6)
List1=[10,20,30,40,50,60,70,80,90,100]
List2=list1[S1]
List3=[1,2,3,4,5,6,7,8,9,10]
List4=list3[S1]
```