Slicing
Slicing is a process reading more than one value/item from sequence.
Reading sub list (sequence) from list (sequence) is called slicing.
This slicing is done in different ways.

1. Slice operator
2. Slice object

This slicing required generating multiple indexes. This multiple indexes are generated using slice operator or slice object.

Syntax of slice operator
Syntax: sequence-name[start:stop:step]

Internally slice operator uses range for generating multiple indexes

Syntax1: sequence-name[::] → start=0,stop=lengthlist,step=1

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

Syntax2: sequence-name[::step]→ if step is +ve start=0,stop=len of list
If step is –ve start=-1,stop=-(len of list+1)

```
>>> list1=[10,20,30,40,50,60,70,80,90,100]
>>> list2=list1[::1]
>>> print(list2)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list3=list1[::2]
>>> print(list3)
[10, 30, 50, 70, 90]
>>> list4=list1[::3]
>>> print(list4)
```

```
[10, 40, 70, 100]
>>> list5=list1[::-1]
>>> print(list5)
[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]
>>> list6=list1[::-2]
>>> print(list6)
[100, 80, 60, 40, 20]
```

```
>>> str1="ABC"
>>> str2=str1[::-1]
>>> print(str1)
ABC
>>> print(str2)
CBA
>>> for value in range(1,11)[::-1]:
...     print(value,end=' ')
...
...
10 9 8 7 6 5 4 3 2 1
>>> for value in range(1,11)[::2]:
...     print(value,end=' ')
...
...
1 3 5 7 9
>>> r1=range(1,11)
>>> print(r1[0])
1
>>> print(r1[-1])
10
```

Syntax-3: sequence-name[start::] → stop=len of lsit,step=1

```
>>> list1=[10,20,30,40,50,60,70,80,90,100]
>>> list2=list1[5:]
>>> print(list2)
[60, 70, 80, 90, 100]
>>> list3=list1[-6:]
>>> print(list3)
[50, 60, 70, 80, 90, 100]
```

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

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

Syntax5: sequence-name[start:stop]

```
>>> list1=[10,20,30,40,50,60,70,80,90,100]
>>> list2=list1[0:5]
>>> print(list2)
[10, 20, 30, 40, 50]
>>> list3=list1[5:8]
>>> print(list3)
[60, 70, 80]
>>> list4=list1[3:-3]
>>> print(list4)
[40, 50, 60, 70]
>>> list5=list1[-3:3]
>>> print(list5)
[]
```

Syntax6: sequence-name[start:stop:step]

```
>>> list1=[10,20,30,40,50,60,70,80,90,100]
>>> list2=list1[3:9:2]
>>> print(list2)
[40, 60, 80]
>>> list3=list1[-3:-9:-1]
>>> print(list3)
[80, 70, 60, 50, 40, 30]
```

```
>>> list4=list1[9:-1:-1]
>>> print(list4)
[]
>>> list4=list1[9::-1]
>>> print(list4)
[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]
>>> list5=list1[-10::1]
>>> print(list5)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list6=list1[3:-3:2]
>>> print(list6)
[40, 60]
```

Slice object

Slice object represents slice operator values (start,stop,step)
Slice object is reusable.

Syntax: slice(stop)
Syntax: slice(start,stop,step)

Example
```
s1=slice(3)
list1=[10,20,30,40,50,60,70,80,90,100]
list2=list1[s1]
print(list1)
print(list2)
```

Output
```
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
[10, 20, 30]
```

Example:
```
s1=slice(2,9,1)
list1=[10,20,30,40,50,60,70,80,90,100]
list2=list1[s1]
print(list1)
print(list2)
```

Output

```
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
[30, 40, 50, 60, 70, 80, 90]
```

What is difference between indexing and slicing?

Indexing is used for reading one item

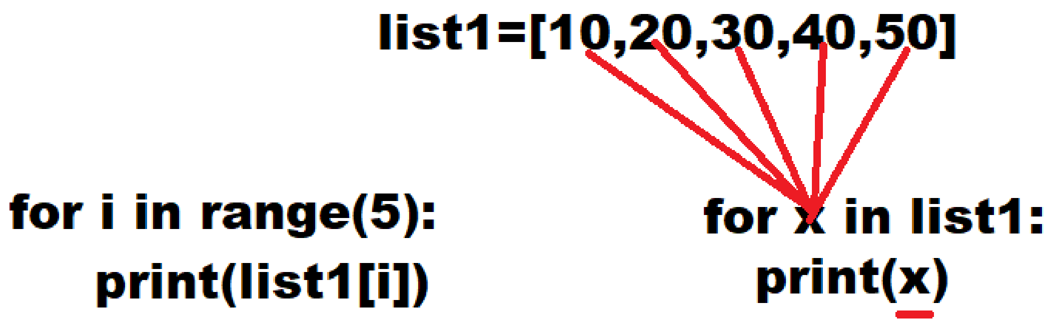
Slicing is used for reading more than one item/value.

for loop

for loop is used to read values from iterables/collections.

Syntax:

```
for variable in iterable:
      statement-1
      statement-2
```



Example	10

list1=[10,20,30,40,50,60,70,80,90,100]
tot=0
for x in list1:
    print(x)
    tot=tot+x
 	
print(f'Total is {tot}')
	20
	30
	40
	50
	60
	70
	80
	90
	100
	Total is 550

Iter()
enumerate()