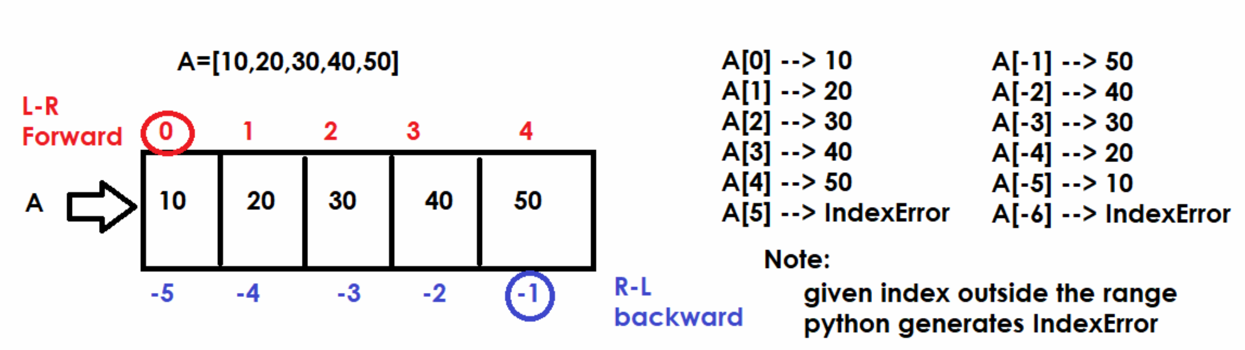
## How to read content of list or sequence?

The content of list or sequence is read in different ways

1. Index
2. Slicing
3. for loop
4. iterator
5. enumerate

## Index

Index an integer value
Each value or element in list or sequence is identified with index.
This index can be +ve or –ve
+ve index starts at 0, to read elements or values from left to right
-ve index starts at -1, to read elements or values from right to left.
This index is used as subscript to read value from sequence



```
A[0] --> 10        A[-1] --> 50
A[1] --> 20        A[-2] --> 40
A[2] --> 30        A[-3] --> 30
A[3] --> 40        A[-4] --> 20
A[4] --> 50        A[-5] --> 10
A[5] --> IndexError   A[-6] --> IndexError
```

Note:
given index outside the range
python generates IndexError

**Example:**

```
A=[10,20,30,40,50]

print(A[0],A[1],A[2],A[3],A[4])
print(A[-1],A[-2],A[-3],A[-4],A[-5])

B=[10,20,30,40,50,60,70,80,90,100]
for i in range(10): # start=0,stop=10,step=1
    print(B[i],end=' ')
```

**Output**

```
10 20 30 40 50
50 40 30 20 10
```

```
10 20 30 40 50 60 70 80 90 100
```

**len():** It is a predefined function in python, this function count of values or length of any collection or iterable.

**Syntax**: len(iterable)

```
>>> A=[10,20,30,40,50]
>>> len(A)
5
>>> B=[]
>>> len(B)
0
```

**Example:**

```
# Given list of integer values/elements count
# even numbers and odd numbers

A=[4,8,1,3,21,22,65,34,12,76,56,43,24,22,12,21,32]
ec=0
oc=0

for i in range(len(A)):
    if A[i]%2==0:
        ec=ec+1
    else:
        oc=oc+1

print(f'List is {A}')
print(f'Count of Even Numbers {ec}')
print(f'Count of Odd Numbers {oc}')
```

**Output**

```
List is [4, 8, 1, 3, 21, 22, 65, 34, 12, 76, 56, 43, 24, 22, 12, 21, 32]
Count of Even Numbers 11
Count of Odd Numbers 6
```

**Example:**

```
# Write a program print sum of elements of list
```

```
A=[10,50,60,30,40,22,44,70,80,66]
s=0

for i in range(len(A)):
    s=s+A[i]

print(f'List is {A}')
print(f'Sum of elements are  {s}')
```

**Output**

```
List is [10, 50, 60, 30, 40, 22, 44, 70, 80, 66]
Sum of elements are  472
```

**Example:**

```
# Write a program to read elements from list
# right to left

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

for i in range(-1,-(len(A)+1),-1):
    print(A[i],end=' ')

# read using +ve index from right to left
print()
for i in range(len(A)-1,-1,-1):
    print(A[i],end=' ')

# read using -ve index from left to right
print()
for i in range(-len(A),0,1):
    print(A[i],end=' ')

print()
for i in range(0,len(A),2):
    print(A[i],end=' ')

print()
for i in range(-1,-(len(A)+1),-2):
    print(A[i],end=' ')
```

**Output**

```
100 90 80 70 60 50 40 30 20 10
100 90 80 70 60 50 40 30 20 10
10 20 30 40 50 60 70 80 90 100
10 30 50 70 90
100 80 60 40 20
```

**Example:**

```
# Write a program to print sum and avg of given list of values

A=[10,60,20,54,89,100,34,56,12,21,34,54,23,22]
s=0

for i in range(len(A)):
    s=s+A[i]

avg=s/len(A)
print(f'List is {A}')
print(f'Sum is {s}')
print(f'Avg is  {avg:.2f}')
```

**Output**

```
List is [10, 60, 20, 54, 89, 100, 34, 56, 12, 21, 34, 54, 23, 22]
Sum is 589
Avg is  42.07
```

**Slicing**

Slicing is an operation of reading multiple values from sequence
Slicing is supported by only sequence data types

**What is difference between indexing and slicing?**

Indexing and slicing are both methods used to access elements within sequences like strings, lists, and tuples in Python, but they serve different purposes. Indexing retrieves a single element, while slicing extracts a subsequence.

Slicing is done in two ways

1. slice operator
2. slice object

slicing generates multiple indexes for reading multiple values.

**Slice operator**

Slice operator required 3 inputs

1. start index
2. stop index
3. step

**Note:** internally slice operator or slice object uses range for generating multiple indexes