How to read data or elements or values from list?

Any sequence data can read in different ways

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

indexing

What is index?

An index integer value

Each value or item in sequence/list is identified with unique number called index.

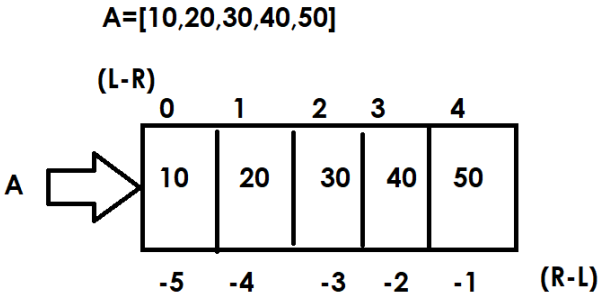
This index can be +ve or –ve

+ve index start at 0, which is used to read data from left to right

-ve index start at -1, which is used to read data from right to left

This index is used as subscript to read and write data in list

Syntax: list-name[index]




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

A[-1] --> 50
A[-2] --> 40
A[-3] --> 30
A[-4] --> 20
A[-5] --> 10
A[-6] --> 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=list(range(10,110,10))
for i in range(10):
    print(B[i],end=' ')

print()
for i in range(-1,-11,-1):
    print(B[i],end=' ')

print()
for i in range(5):
    print(B[i],end=' ')

print()
for i in range(-5,0,1):
    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
100 90 80 70 60 50 40 30 20 10
10 20 30 40 50
60 70 80 90 100
```

Example:

```
# Write a program to find sum,avg of values in the following
# list

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

```
s=0

for i in range(5):
    s=s+A[i]

print(f'List is {A}')
print(f'Sum is {s}')
a=s/5
print(f'Avg is {a:.2f}')
```

Output

List is [10, 20, 30, 40, 50]
Sum is 150
Avg is 30.00

len() : It is a predefined function in python, this function returns length of collection (OR) count of values/items

```
len(iterable/collection)
```

Example:

```
# Write a program to find minimum,maximum value
# of given list

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

max_value=A[0]
min_value=A[0]
for i in range(len(A)):
    if A[i]>max_value:
        max_value=A[i]
```

```
    elif A[i]<min_value:
        min_value=A[i]

print(f'List is {A}')
print(f'Maximum Value {max_value}')
print(f'Minimum Value {min_value}')
```

Output

```
List is [40, 20, 10, 70, 30, 60, 50]
Maximum Value 70
Minimum Value 10
```

Example:

```
# Write a program to count +ve,-ve and zeros in given list
A=[10,20,-40,40,-50,0,-70,0,90,-4,0,0]
c1=0
c2=0
c3=0
for i in range(len(A)):
    if A[i]>0:
        c1+=1
    elif A[i]<0:
        c2+=1
    else:
        c3+=1

print(f'List is {A}')
print(f'+ve Count {c1}')
print(f'-ve Count {c2}')
print(f'zero Count {c3}')
```

Output

List is [10, 20, -40, 40, -50, 0, -70, 0, 90, -4, 0, 0]
+ve Count 4
-ve Count 4
zero Count 4

Example:

```
# Write a program to count even and odd numbers in given
# list of numbers

A=[1,6,9,12,25,56,76,32,33,54,75,89,67,57,35,4,5,2]
ec=0
oc=0
for i in range(len(A)):
    if A[i]%2==0:
        ec+=1
    else:
        oc+=1
print(f'List is {A}')
print(f'Even Numbers Count {ec}')
print(f'Odd Numbers Count {oc}')
```

Output

List is [1, 6, 9, 12, 25, 56, 76, 32, 33, 54, 75, 89, 67, 57, 35, 4, 5, 2]
Even Numbers Count 8
Odd Numbers Count 10

Example:

```
>>> r1=range(10,60,10)
>>> r1[0]
10
>>> r1[-1]
50
```

```
>>> r1[1]
20
```

Slicing

Slicing is a process of reading more than one value from sequence.
Slicing returns sub sequence from sequence.
Slicing generates multiple indexes for reading multiple values.
Slicing is done in two ways

1. using slice operator
2. using slice object

Using slice operator

Slicing operator generates multiple indexes for reading multiple values. These multiple indexes are generated internally using range

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

Slice operator required 3 inputs

1. start starting index (included)
2. stop stop index (excluded)
3. step increment/decrement

Example:

```
A=list(range(10,110,10))
print(A)
B=A[0:5:1]
print(B)
C=A[-5::1]
print(C)
D=A[0:10:2]
print(D)
E=A[1:10:2]
```

```
print(E)
F=A[-1:-11:-1]
print(F)
G=A[::] # start=0,stop=len of sequence,step=1
print(G)
H=A[:]
print(H)
I=A[::2] # start=0,stop=len of sequence,step=2
print(I)
H=A[::-1] # start=-1,stop=-len of sequence,step=-1
print(H)
K=A[::-2] # start=-1,stop=-len if sequence,step=-2
print(K)
L=A[1::-2] # start=-1
print(L)
L=A[3::-2] # start=-3
print(L)
```

Output

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

Example:

```
>>> r1=range(10,110,10)
>>> r2=r1[0:5]
>>> print(r1)
range(10, 110, 10)
>>> print(r2)
range(10, 60, 10)
```