Sequences
Sequence data types allow organizing data in memory in sequential order. By storing in sequential order, the data can be read and write sequential and randomly.
Sequences are index based collection.
In application development, sequences are used to group individual objects and allow duplicates and reading and writing is done sequential or random.

List
List is a sequence data type.
List is a mutable sequence data type. After creating list changes can be done.
Lists are mutable sequences, typically used to store collections of homogeneous items and heterogeneous items.
List allows duplicate items or elements or values.

How to create list?

Lists may be constructed in several ways:

1. Using a pair of square brackets to denote the empty list: []
2. Using square brackets, separating items with commas: [a], [a, b, c]
3. Using a list comprehension: [x for x in iterable]
4. Using the type constructor or function: list() or list(iterable)

Note: **All** collections are called iterables. Iterable types allows to read individual values/elements.

Creating empty list

```
>>> list1=[]
>>> print(list1)
[]
>>> print(type(list1))
<class 'list'>
```

Creating list with values/items/elements.

Using square brackets, separating items with commas: [a], [a, b, c]

Example

```
>>> list2=[1,2,3,4,5]
>>> print(list2)
[1, 2, 3, 4, 5]
>>> list3=[1.5,2.5,3.5,1.8,2.8]
>>> print(list3)
[1.5, 2.5, 3.5, 1.8, 2.8]
>>> list4=["java","python","oracle",".net"]
>>> print(list4)
['java', 'python', 'oracle', '.net']
>>> student=[101,"naresh","python",4000.0]
>>> print(student)
[101, 'naresh', 'python', 4000.0]
>>> list5=[1,1,1,1,1,1,2,3,4,2,6,8,9,3,2,1,1]
>>> print(list5)
[1, 1, 1, 1, 1, 1, 2, 3, 4, 2, 6, 8, 9, 3, 2, 1, 1]
>>> list6=[10,20,30,40,50]
>>> print(list6)
[10, 20, 30, 40, 50]
>>> s1={10,20,30,40,50}
>>> print(s1)
{50, 20, 40, 10, 30}
```

What is iterable in Python?

Definition: An iterable is any Python object capable of returning its members one at a time, permitting it to be iterated over in a for-loop. Familiar examples of iterables include lists, tuples, and strings - any such sequence can be iterated over in a for-loop.

list() function

This function is used to convert other iterables into list type.
Syntax-1: list()
Syntax-2: list(iterable)
list() → This create empty list
list(iterable) → This converts existing iterables into list type

Example:

```
>>> list7=list()
>>> print(list7)
[]
>>> list8=list(range(1,6))
```

```
>>> print(list8)
[1, 2, 3, 4, 5]
>>> list9=list(range(10,60,10))
>>> print(list9)
[10, 20, 30, 40, 50]
>>> list10=list(range(50,0,-10))
>>> print(list10)
[50, 40, 30, 20, 10]
>>> list11=list("PYTHON")
>>> print(list11)
['P', 'Y', 'T', 'H', 'O', 'N']
>>> list12=list(list10)
>>> print(list12)
[50, 40, 30, 20, 10]
```

Reading content of list

Python allows reading content of list/sequences in different ways.

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

Using Index

What is index?

Index is an integer value used to identify each value in sequence (OR) each location in sequence.

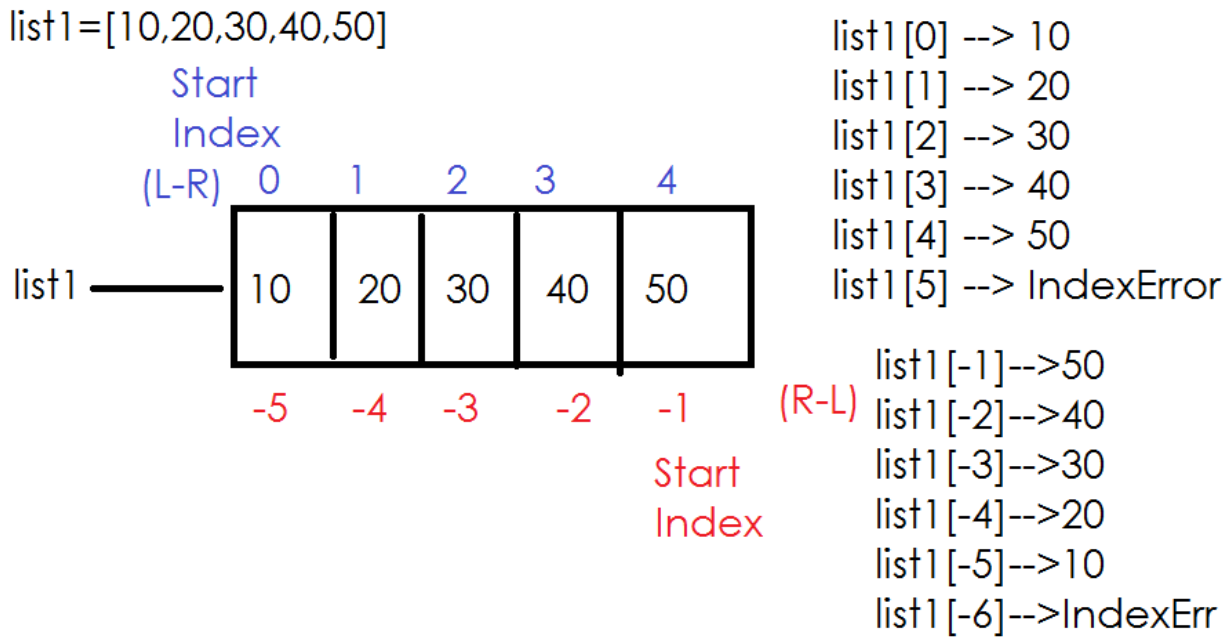
Using index programmer can read one value. This index can be +ve or –ve.

+ve index start at 0 (Left-Right)

-ve index start at -1 (Right-Left)

Index allow to read in sequential and random.

Syntax: list-name[index]



Example

```
>>> list1=[10,20,30,40,50]
>>> print(list1[0],list1[1],list1[2],list1[3],list1[4])
10 20 30 40 50
>>> print(list1[-1],list1[-2],list1[-3],list1[-4],list1[-5])
50 40 30 20 10
>>> r1=range(10,60,10)
>>> r1[0]
10
>>> r1[1]
20
>>> r1[-1]
50
```

Example:

```
# Create list with 5 values and read each value

list1=[1,2,3,4,5]

print("Reading data from L-R")
for i in range(5): # start=0,stop=5,step=1 -0 1 2 3 4
    print(list1[i])

print("Reading data from R-L")
```

```
for i in range(-1,-6,-1):#start=-1,stop=-6,step=-1
    print(list1[i])

for i in range(0,5,2):
    print(list1[i])

for i in range(-1,-6,-2):
    print(list1[i])
```

Output:

```
Reading data from L-R
1
2
3
4
5
Reading data from R-L
5
4
3
2
1
1
3
5
5
3
1
```

Example:

```
# Create a list with 5 values
# count even numbers and odd number

list1=[5,2,7,4,8]
ec=0
oc=0
for i in range(len(list1)): # start=0,stop=5,step=1
    if list1[i]%2==0:
        ec+=1
    else:
```

```
        oc+=1


print(f'List is {list1}')
print(f'Even Numbers Count {ec}')
print(f'Odd Numbers Count {oc}')
```

Output:

```
List is [5, 2, 7, 4, 8]
Even Numbers Count 3
Odd Numbers Count 2
```