LIST - DATA STRUCTURE:

- A list is a collection of elements. These elements may be homogeneous(same types) or heterogeneous (diffrent types).
- A list also allows duplicate elements.
- Insertion order is preserved in list.
- List elements are separated by commas and enclosed within square brackets [10,20,'abc'].
- Every element in the list has its own unique index number.
- List supports both forward indexing and backward indexing, forward index starts from 0 and backward index starts from -1.
- We access either specific element by using "indexing" or set of elements by using "slicing" from the List.
  For example, l[0] --->> indexing l[0:3] ----->> slicing
- We can create list in different ways.
  a. By using "list()" function,
  b. By using square brackets []
  c. By using range() function.
- List objects are mutable means we can change the list elements.

Creating List by using list():

1. This list() allows only one string value with set of characters.
2. If we give int type data in the list() function then interpreter will throw '**TypeError'** error.

Example:

```
>>> List1 =  list()          #creating empty list
>>> print(List1)             []
>>> type(List1)                    <class 'list'>
>>> List1  =  list('python')     #creating list with set of characters
>>> print(List1)             ['p', 'y', 't', 'h', 'o', 'n']
>>> type(List1)                    <class 'list'>
```

Creating list by using square brackets []

```
>>> List1  =  []                          #creating empty list
```

```
>>> print(List1)                    []
>>> type(List1)                     <class 'list'>
>>> List1  =  [1,2,3,4,5]           #creating list with homogeneous elements
>>> print(List1)                    [1, 2, 3, 4, 5]
>>> type(List1)                     <class 'list'>
>>> List1 = [10,11,'Python',5.5,True,2+3j]    #creating list with heterogeneous
elements
>>> print(List1)                          [10, 11, 'Python', 5.5, True, (2+3j)]
>>> type(List1)                           <class 'list'>
```

Creating list by using range() function:

We can also use range() function to create list.

Syntax : range(StartingIndexValue, LastValue-1, RangeValue)

For example: range(10)

a. Here, both StartingIndexValue and RangeValue are optional.

b. The default StartingIndexValue is 0.

c. The default RangeValue is 1.

Example:

```
>>> List1  =  list(range(10))
>>> print(List1)              [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> type(List1)                     <class 'list'>
>>> List1  =  list(range(0,10))
>>> print(List1)              [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> type(List1)                     <class 'list'>
>>> List1  =  list(range(2,8))
>>> print(List1)              [2, 3, 4, 5, 6, 7]
>>> type(List1)                     <class 'list'>
```

List Indexing

1. By using list indexing we can fetch specific element from the list.
2. It supports both forward and backward indexing

```
       0    1      2        3           4         5          6
List1 =[ 10,   20,     30,     'Python',      True,      1.5,      2+3j]
      -7    -6      -5       -4          -3       -2         -1
```

Example:

```
>>> List1  =  [10,20,30,'Python',True,1.5,2+3j]
>>> print(List1[0])                    10
>>> print(List1[1])                    20
```

List Slicing

1.By using slicing we can fetch set of characters from list.

2.It also supports both forward and backward indexing

3.Colon (:) is the slicing operator.

Example:

```
>>> List1  =  [10,20,30,'Python',True,1.5,2+3j]
>>> print(List1[0:2])                  [10, 20]
>>> print(List1[2:5])                  [30, 'Python', True]
```

NOTE : List is a "mutable" object that means we can update or replace the existing list with new elements. But id reference value is not changed.

List Concatenation:

Python supports concatenating two or more lists into single list.

Example:

```
>>> list1 =  [10,20,50]
>>> list2 =  [70,10,20,50]
>>> list1 + list2                  #  [10, 20, 50, 70, 10, 20, 60]
```

List Multiplication or List Repetition

Python supports multiplying the given list into N number of times.

Example:

```
>>> List1  =  [10,'Python',5.5]
>>> List2 =  List1*3
>>> print(List2)   # [10, 'Python', 5.5, 10, 'Python', 5.5, 10, 'Python', 5.5]
```