---: List :---

Whenever we want to create a group of objects where we want below mention properties, then we are using list sequence.

1. Duplicates are allowed.
2. Order is preserved.
3. Objects are mutable.
4. Indexing are allowed.
5. Slicing are allowed.
6. Represented in square bracket with comma separated objects.
7. Homogeneous and Heterogeneous both objects are allowed.

1. Duplicates are allowed.

```
List=['neeraj', 10,20,30,10,20]
print(List)

O/P:--
['neeraj', 10, 20, 30, 10, 20]
```

2. Order is preserved:

```
List=['neeraj', 10,20,30,10,20]
x=0
for i in List:
    print('List[{}] = '.format(x),i)
    x=x+1

O/P:--
List[0] =  neeraj
List[1] =  10
List[2] =  20
List[3] =  30
List[4] =  10
List[5] =  20
```

3. Objects are mutable.

```
List=['neeraj', 10,20,30,10,20]
x=0
for i in List:
    print('List[{}] = '.format(x),i)
```

```
   x=x+1
List[0]="Arvind"
print(List)

O/P:--
List[0] =  neeraj
List[1] =  10
List[2] =  20
List[3] =  30
List[4] =  10
List[5] =  20
['Arvind', 10, 20, 30, 10, 20]
```

4. Indexing are allowed.

```
List=['neeraj', 10,20,30,10,20]
print(List[0])
print(List[1])
print(List[2])
print(List[3])
print(List[4])
print(List[5])

O/P:--
neeraj
10
20
30
10
20
```

5. Slicing are allowed:

```
List=['neeraj', 10,20,30,10,20]
print(List[:5])

O/P:--
['neeraj', 10, 20, 30, 10]
```

```
List=['neeraj', 10,20,30,10,20]
print(List[::-1])

O/P:--
[20, 10, 30, 20, 10, 'neeraj']
```

Inbuilt functions in list:

1.len(list)
2.max(list) - homogeneous collection required
3.min(list) - homogeneous collection required
4.sum(list) - integer homogeneous collection required
5.list(tuple)
6.type(list)
7.id()
8.list()

Methos:--

1. **list.append(obj/list/str)-** add object in last

```
animals = ['cat', 'dog', 'rabbit']
# Add 'rat' to the list
animals.append('rat')
print('Updated animals list: ', animals)

O/P:--
Updated animals list:  ['cat', 'dog', 'rabbit', 'rat']

animals = ['cat', 'dog', 'rabbit']
wild_animals = ['tiger', 'fox']
animals.append(wild_animals)
print('Updated animals list: ', animals)

O/P:--
Updated animals list:  ['cat', 'dog', 'rabbit', ['tiger', 'fox']]
```

2. **list.count(obj)** – count how many times given-object are present in list

```
numbers = [2, 3, 5, 2, 11, 2, 7]
count = numbers.count(2)
```

```
print('Count of 2:', count)

O/P:--
Count of 2: 3

# vowels list
vowels = ['a', 'e', 'i', 'o', 'i', 'u']
count = vowels.count('i')
print('The count of i is:', count)
count = vowels.count('p')
print('The count of p is:', count)

O/P:--
The count of i is: 2
The count of p is: 0
# random list
random = ['a', ('a', 'b'), ('a', 'b'), [3, 4]]
count = random.count(('a', 'b'))
print("The count of ('a', 'b') is:", count)
count = random.count([3, 4])
print("The count of [3, 4] is:", count)

O/P:--
The count of ('a', 'b') is: 2
The count of [3, 4] is: 1
```

3. **list.extend(list1)** – add list1 in last of list.

```
# create a list
list1 = [2, 3, 5]
list2 = [1, 4]
list1.extend(list2)
print('List after extend():', list1)

O/P:--
List after extend(): [2, 3, 5, 1, 4]


list = ['Hindi']
tuple = ('Spanish', 'English')
```

```
set = {'Chinese', 'Japanese'}
list.extend(tuple)
print('New Language List:', list)
list.extend(set)
print('Newer Languages List:', list)

O/P:--
New Language List: ['Hindi', 'Spanish', 'English']
Newer Languages List: ['Hindi', 'Spanish', 'English', 'Japanese', 'Chinese']
```

4. **list.insert(index,obj)** – insert given object in given index.
5. **list.pop()** – delete bydefault last object from given list.
6. **list.remove(obj)** – Remove given object from given list.
7. **list.reverse() –**

```
Example:---
numbers = ['Neeraj',2, 3, 5, 7]
numbers.reverse()
print('Reversed List:', numbers)

O/P:--
Reversed List: [7, 5, 3, 2, 'Neeraj']

Example:----
numbers = ['Neeraj',2, 3, 5, 7]
print(numbers[::-1])

O/P:--
[7, 5, 3, 2, 'Neeraj']

Example:----
numbers = ['Neeraj',2, 3, 5, 7]
# print(numbers[::-1])
list=[]
for i in reversed(numbers):
    list.append(i)
print(list)

O/P:--
[7, 5, 3, 2, 'Neeraj']
```

8. **list.sort**(reverse=True/False) default-False

```
Example:---
numbers = [2, 3, 7, 5, 4]
numbers.sort()
print('Sort_List:', numbers)

O/P:--
Sort_List: [2, 3, 4, 5, 7]
Example:---
numbers = [2, 3, 7, 5, 4]
numbers.sort(reverse=True)
print('Sort_List:', numbers)

O/P:--
Sort_List: [7, 5, 4, 3, 2]
```

List-related operators

1. **Concatenation Operator (+):---**

```
a = [1, 2, 3]
b = [4, 5]
print(a + b)

O/P:--
[1, 2, 3, 4, 5]
```

2. **Repetition Operator (*) :----**

```
lst = [1, 2]
print(lst * 3)

O/P:---
[1, 2, 1, 2, 1, 2]
```

3.**Membership Operators (in, not in) :-----**

```
a = [10, 20, 30]
```

```
print(20 in a)
print(100 not in a)

O/P:---
True
True
```

4. **Comparison Operators (==, !=, <, >, <=, >=) :----**

```
a = [1, 2, 3]
b = [1, 2, 3]
c = [1, 3, 2]

print(a == b)
print(a < c)

O/P:---
True
True
```

5. Indexing Operator ([])

```
nums = [100, 200, 300]
print(nums[0])
print(nums[-1])

O/P:--
100
300
```

6. Slicing Operator (**[:]**)

```
nums = [10, 20, 30, 40, 50]
print(nums[1:4])
print(nums[:3])
print(nums[::2])

O/P:---
[20, 30, 40]
```

```
[10, 20, 30]
[10, 30, 50]
```

7. Assignment Operators (=, +=, *=)

```
lst = [1, 2]
lst += [3, 4]
print(lst)
lst *= 2
print(lst)

O/P:---
[1, 2, 3, 4]
[1, 2, 3, 4, 1, 2, 3, 4]
```

8. Identity Operators (is, is not)

```
a = [1, 2, 3]
b = [1, 2, 3]
c = a

print(a == b)
print(a is b)
print(a is c)

O/P:---
True
False
True
```

9. Logical Operators (**and**, **or**, **not**)

```
a = [1]
b = []

print(a and b)
print(a or b)
print(not a)
print(not b)
```

```
O/P:---
[]
[1]
False
True
```

9. Comparison Using **any()** and **all()** (Indirect use)

```
lst = [True, False, True]
print(all(lst))
print(any(lst))

O/P:---
False
True
```