

Chapter : 8 | List

1. What is a List ?

A List in Python is a collection of ordered, mutable (changeable) items, enclosed in square brackets [], and items can be of different data types.

Code -

```
# Example of a list
fruits = ['apple', 'banana', 'cherry']
```

2. Declaring / Creating / Initializing Lists

Lists can be :

- Empty
- Initialized with elements
- Mixed data types allowed

Code -

```
empty_list = []
num_list = [1, 2, 3, 4]
mixed_list = [1, 'hello', 3.14, True]
```

3. Accessing List Elements (Indexing)

List elements are accessed using indexes, starting from 0.

Code -

```
fruits = ['apple', 'banana', 'cherry']
print(fruits[0]) # apple
print(fruits[-1]) # cherry (negative indexing from end)
```

4. Traversing a List

Using a for loop to access all items.

Code -

```
for fruit in fruits:
    print(fruit)
```

Using index-based traversal:

Code -

```
for i in range(len(fruits)):
    print(fruits[i])
```

5. Aliasing

Aliasing means two variables refer to the same list object.

Code -

```
a = [1, 2, 3]
b = a      # b is alias of a
b[0] = 10
print(a)   # [10, 2, 3] - changes reflect in both
```

6. Comparing Lists

Two lists are equal if all elements and their order are the same.

Code -

```
a = [1, 2, 3]
b = [1, 2, 3]
c = [3, 2, 1]
print(a == b) # True
print(a == c) # False
```

7. Operations on Lists

<ul style="list-style-type: none">◆ Concatenation Using + to join two lists. Code - <pre>a = [1, 2] b = [3, 4] print(a + b) # [1, 2, 3, 4]</pre>	<ul style="list-style-type: none">◆ Repetition / Replication Using * to repeat list. Code - <pre>a = [0] print(a * 4) # [0, 0, 0, 0]</pre>
<ul style="list-style-type: none">◆ Membership Testing Using in / not in to test presence. Code - <pre>fruits = ['apple', 'banana'] print('apple' in fruits) # True print('mango' not in fruits) # True</pre>	<ul style="list-style-type: none">◆ Indexing Accessing elements with position. Code - <pre>print(fruits[1]) # banana</pre>
<ul style="list-style-type: none">◆ Slicing Accessing parts of list: list[start:stop:step] Code - <pre>nums = [10, 20, 30, 40, 50] print(nums[1:4]) # [20, 30, 40] print(nums[:3]) # [10, 20, 30] print(nums[::-1]) # [50, 40, 30, 20, 10] (reverse)</pre>	

8. Nested Lists

Lists within lists.

Code -

```
nested = [[1, 2], [3, 4]]
print(nested[0])      # [1, 2]
print(nested[0][1])   # 2
```

9. Copying Lists

- ◆ Shallow Copy (Using slicing or list())

Code -

```
a = [1, 2, 3]
b = a[:] # or list(a)
b[0] = 10
print(a) # [1, 2, 3] - no change
```

10. Built-in Functions and Methods

<ul style="list-style-type: none">◆ append(x) Adds a single item at the end. Code - <pre>lst = [1, 2] lst.append(3) print(lst) # [1, 2, 3]</pre>	<ul style="list-style-type: none">◆ extend(iterable) Adds all items from another list. Code - <pre>lst = [1, 2] lst.extend([3, 4]) print(lst) # [1, 2, 3, 4]</pre>
<ul style="list-style-type: none">◆ insert(index, value) Inserts value at specified index. Code - <pre>lst = [1, 2] lst.insert(1, 'a') print(lst) # [1, 'a', 2]</pre>	<ul style="list-style-type: none">◆ Updating List Elements Change using indexing. Code - <pre>lst = [1, 2, 3] lst[1] = 20 print(lst) # [1, 20, 3]</pre>
<ul style="list-style-type: none">◆ len(list) Returns the number of elements. Code - <pre>len([1, 2, 3]) # 3</pre>	<ul style="list-style-type: none">◆ sort() Sorts the list in-place (ascending by default). Code - <pre>nums = [3, 1, 2] nums.sort() print(nums) # [1, 2, 3]</pre>
<ul style="list-style-type: none">◆ clear() Removes all elements. Code - <pre>lst = [1, 2] lst.clear()</pre>	<ul style="list-style-type: none">◆ count(x) Counts occurrences of x. Code - <pre>lst = [1, 2, 2, 3] print(lst.count(2)) # 2</pre>

<pre>print(lst) # []</pre>	
----------------------------	--

11. Deletion Operations

♦ del statement

Delete element or slice.

Code -

```
a = [1, 2, 3]
del a[1]
print(a) # [1, 3]
del a[:] # delete all
print(a) # []
```

♦ pop(index)

Removes and returns element at index. Default is last.

Code -

```
a = [10, 20, 30]
print(a.pop()) # 30
print(a) # [10, 20]
```

♦ remove(x)

Removes first occurrence of x.

Code -

```
a = [1, 2, 3, 2]
a.remove(2)
print(a) # [1, 3, 2]
```

FundaClass Academy
Assignment Series