Introduction to Lists in Python

1. What is a List in Python?

A `list` in Python is a built-in data type used to store collections of items. Lists are:

- ① Ordered
- ② Mutable (can be changed)
- ③ Allow duplicate elements

Lists are defined using square brackets `[]`.

2. Creating a List

```
# A list of numbers
numbers = [1, 2, 3, 4, 5]

# A list of strings
fruits = ["apple", "banana", "cherry"]
```

1st 2nd 3rd 4th 5th

1st 2nd 3rd

← There are 3 elements

3. Accessing List Elements

Elements in a list are accessed using index notation. Indexing starts at 0.

```
print(fruits[0])    # Output: apple

print(numbers[2])   # Output: 3
```

frits = ["apple", "banana", "Cherry"]
1st 2nd 3rd

4. Modifying a List

```
fruits[1] = "orange"
print(fruits)  # Output: ['apple', 'orange', 'cherry']
```
✓

5. List Operations

- 1 **Append:** Add an item to the end ⇒ `fruits.append("mango")` index
- 2 **Insert:** Add an item at a specific index ⇒ `fruits.insert(1, "grape")`
- 3 **Remove:** Remove an item by value ⇒ `fruits.remove("apple")` item to be removed
- 4 **Pop:** Remove the last item (or by index) ⇒ `last_item = fruits.pop()`

6. Looping Through a List

fruits = ["apple", "orange", "cherry"]

```
for fruit in fruits:
    print("I like", fruit)
```

I like apple

I like orange

I like cherry

7. List Slicing

Slicing is used to get a part (subset) of the list.

```
# Get first two elements
print(fruits[0:2])
```

apple, orange

8. Importance of Lists

- 1 **Flexible Data Structure:** Lists can store any type of data.
- 2 **Dynamic Size:** No need to declare size ahead of time.
- 3 **Built-in Methods:** Python provides many convenient list methods.
- 4 **Widely Used:** Lists are one of the most used data structures in Python.

9. Drawbacks of Lists

- 1 **Slower for Large Data:** Lists are not optimized for performance with huge datasets.
- 2 **Inefficient Insertion/Deletion:** Inserting or deleting in the middle is slower than arrays in some languages.
- 3 **Memory Usage:** Lists use more memory because of their dynamic nature.

10. Summary

Lists in Python are versatile, dynamic, and easy to use. They are ideal for most small to medium-scale data storage and manipulation tasks. However, for performance-critical or large-scale tasks, other structures like NumPy arrays or pandas DataFrames may be preferred.