🔹 1. Immutable vs Mutable

Immutable = cannot be changed after creation.
Example: numbers, strings, tuples.

```
x = 5
x = 6   # Here Python created a NEW memory reference for 6, not changing 5
```

-

Mutable = can be changed after creation.
Example: lists, dictionaries, sets.

```
fruits = ["apple", "banana"]
fruits.append("mango")   # Changed the same list, no new memory reference
```

-

🔹 2. Lists in Python

- A **list** is a collection of items (mutable).
- Defined using **square brackets** `[ ]`.

Example:

```
ingredients = ["water", "milk", "black tea"]
```

🔹 3. Adding Items

Append → always adds at the **end**:

```
ingredients.append("sugar")
print(ingredients)  # ['water', 'milk', 'black tea', 'sugar']
```

-

Insert → adds at a specific position (index):

```
ingredients.insert(1, "ginger")
print(ingredients)  # ['water', 'ginger', 'milk', 'black tea',
'sugar']
```

-

Extend → combine two lists:

```
spices = ["ginger", "cardamom"]
ingredients.extend(spices)
print(ingredients)
# ['water', 'milk', 'black tea', 'sugar', 'ginger', 'cardamom']
```

-

🔹 4. Removing Items

Remove by value:

```
ingredients.remove("water")
print(ingredients)  # ['milk', 'black tea', 'sugar']
```

-

Pop by index:

```
ingredients.pop(1)
print(ingredients)  # Removes the item at index 1
```

-

Clear (delete everything):

```
ingredients.clear()
print(ingredients)  # []
```

-

🔹 5. Indexing (positions in a list)

Python counts from **0**.

```
ingredients = ["water", "milk", "ginger"]
print(ingredients[0])  # water
print(ingredients[1])  # milk
print(ingredients[2])  # ginger
```

-

✅ **Summary**:

- **Immutable**: can't be changed (numbers, strings, tuples).
- **Mutable**: can be changed (lists, dicts, sets).
- Lists let you **add, remove, insert, extend, reorder** items freely.
- Lists = Python's version of **arrays** in other languages.