PYTHON : DAY-7

LISTS :

A **list** in Python is:

✔️ Ordered (items have fixed positions)

✔️ Mutable (can be changed after creation)

✔️ Allows duplicates

✔️ Can store mixed data types

EXAMPLE:

```
my_list = ["string", 25, True, 3.5]
```

Creating a list:

Two common ways:

```
lst = list()
lst = []
```

Both create an empty list.

Example lists:

```
fruits = ["banana", "orange", "lemon"]
vegetables = ["tomato", "potato", "cabbage", "onion"]
fruits = ["banana", "orange", "lemon"]
vegetables = ["tomato", "potato", "cabbage", "onion"]
```

`len()` gives number of items

```
len(fruits)        # 3
len(vegetables)    # 4
```

List indexing :

Python supports:

Type	Starts At
Positive Index	0
Negative Index	-1

- Positive indexing :

```
names = ["dermin","cool","notcool"]
names[0] # 'dermin'
```

- Negative indexing :

```
names[-1] # 'notcool'
```

Useful for accessing end of lists.

List Unpacking :

Python allows extracting items into variables

```
x, y, z = ["a","b","c"]
```

With the star operator:

```
a, b, *rest, last = [1,2,3,4,5,6,7]
rest # [3,4,5,6]
```

List slicing :

Syntax: `list[start:end]` (end excluded)

- Positive index :

```
fruits = ["banana", "orange", "mango", "lemon"]
orange_and_mango = fruits[1:3]    # ['orange', 'mango']
print(orange_and_mango)
```

- Negative index:

```
fruits[-3:-1]  # ['orange','mango']
```

- **Common slice shortcuts:**

Expression	Meaning
`lst[:]`	copy list
`lst[:n]`	first n items
`lst[-n:]`	last n items
`lst[::-1]`	reversed list

Modifying Lists:

Lists are mutable:

```
fruits[0] = "avocado"
```

Membership Test:

```
"banana" in fruits   # True
"apple" in fruits    # False
```

Adding Items:

`append()` → **adds at end:**

```
names.append("derminiscool")
```

`insert(index, item)` → **adds at specific position:**

```
names.insert(1, "mid")
```

Removing Items :

Method	Usage	Removes
`remove(x)`	by value	first occurrence
`pop()`	last item	returns removed
`pop(i)`	index	returns removed
`del list[i]`	delete by index	no return
`del list`	delete entire list	destroys object

Example :

```
names.remove("dermin")
names.pop()
names.pop(0)
del names[1]
```

Copying Lists:

```
copy_list = original.copy()
```

Joining Lists:

Using `+`

```
combined = a + b
```

Using `extend()`

```
a.extend(b)
```

Counting Items:

```
names.count("dermin")
```

Finding Index:

```
fruits.index("orange")   # 1
```

Raises error if not found

Reversing a List:

```
fruits.reverse()
```

Also possible using slicing:

```
fruits[::-1]
```

Sorting Lists:

In-place sort

```
names.sort()# ascending
```

```
names.sort(reverse=True)# descending
```

Using `sorted()` (does not modify original)

```
sorted(fruits)
sorted(fruits, reverse=True)
```

Summary :

Feature	Value
Ordered	Yes
Mutable	Yes
Stores duplicates	Yes
Stores mixed types	Yes
Heterogeneous	Supported