

PYTHON LISTS – NOTES

List Modification, Properties and Quick Practice Programs

1. Lists Can Be Modified

Python lists are mutable.

Mutable means that we can change the objects stored in a list after the list has been created.

```
numbers = [10, 20, 30, 40, 50]
numbers[2] = 100
print(numbers)
Output: [10, 20, 100, 40, 50]
```

Originally: [10, 20, 30, 40, 50]

After changing the object at index 2: [10, 20, 100, 40, 50]

Explanation

The third object has been changed from 30 to 100.

2. Adding an Object to a List

We can add a new object to the end of a list using `append()`.

```
numbers = [10, 20, 30]
numbers.append(40)
print(numbers)
Output: [10, 20, 30, 40]
```

Before `append()`: [10, 20, 30]

After `append()`: [10, 20, 30, 40]

Explanation

The object 40 has been added to the end.

3. Removing an Object from a List

We can remove an object using `remove()`.

```
numbers = [10, 20, 30, 40, 50]
numbers.remove(30)
print(numbers)
Output: [10, 20, 40, 50]
```

Explanation

The object 30 has been removed.

4. Changing Multiple Objects

Because lists are mutable, we can change their objects.

```
marks = [50, 60, 70, 80, 90]
marks[0] = 55
marks[4] = 95
print(marks)
Output: [55, 60, 70, 80, 95]
```

5. An Empty List

A list does not have to contain objects when it is created.

```
numbers = []
print(numbers)
```

```
Output: []
```

We can later add objects:

```
numbers = []
numbers.append(10)
numbers.append(20)
numbers.append(30)
print(numbers)
Output: [10, 20, 30]
```

6. Important Properties of Python Lists

We can summarize the basic properties of lists as follows:

1. Lists store multiple objects

A list can store multiple objects.

```
numbers = [10, 20, 30, 40, 50]
```

2. Lists are ordered

The objects have a specific order.

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

3. Lists use square brackets

Lists are written using square brackets [].

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

4. Lists allow duplicate objects

The same value can appear multiple times.

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

5. Lists can contain different types of objects

A list can contain different types of objects.

```
data = [10, 3.14, "Hello", True]
```

6. Lists are mutable

Objects can be changed, added, or removed.

```
numbers = [10, 20, 30]
numbers[1] = 50
```

7. Lists use zero-based indexing

The first object has index 0.

```
numbers = [10, 20, 30]
print(numbers[0])
```

7. A Simple Real-World Example

Suppose we want to store the marks of five students:

```
marks = [85, 72, 91, 68, 77]

print(marks)
print(marks[0])
```

```
print(marks[2])
print(len(marks))
Output: [85, 72, 91, 68, 77]
85
91
5
```

What this example shows

The list stores five marks. marks[0] accesses the first mark, marks[2] accesses the third mark, and len(marks) gives the number of objects.

8. Quick Practice Programs

Try running these programs yourself.

Practice 1

```
numbers = [5, 10, 15, 20, 25]
print(numbers)
```

Practice 2

```
names = ["Amit", "Ravi", "John", "Ali", "David"]
print(names[0])
print(names[4])
```

Practice 3

```
data = [100, 25.5, "Python", True, 200]
print(data)
print(len(data))
```

Practice 4

```
numbers = [10, 20, 30, 40, 50]
numbers[1] = 200
print(numbers)
```

Practice 5

```
numbers = [10, 20, 30]
numbers.append(40)
print(numbers)
```

Practice 6

```
numbers = [10, 20, 30, 40]
numbers.remove(20)
print(numbers)
```

Quick Revision

Python lists are ordered and mutable. Their objects can be accessed using indexes, changed, added, or removed. Lists can contain duplicate objects and objects of different types.

EngineersTutor.com

YouTube: Engineers Tutor