



TOPSTechnologies

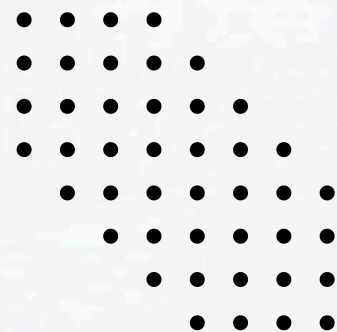
# List Operations

**Presented for :**

TOPs Technologies

**Presented by :**

Sumit B Yadav



## *Que 1*

### 1. Concatenation

Combining two or more lists into a single list using the + operator.

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
result = list1 + list2
print(result) # Output: [1, 2, 3, 4, 5, 6]
```

### 2. Repetition

Creating a new list by repeating the elements of an existing list a specified number of times using the \* operator.

```
list1 = [1, 2, 3]
result = list1 * 3
print(result) # Output: [1, 2, 3, 1, 2, 3, 1, 2, 3]
```

### 3. Membership

Checking if an element exists in a list using the in or not in operators.

```
list1 = [1, 2, 3, 4, 5]
print(3 in list1)    # Output: True
print(6 in list1)    # Output: False
print(6 not in list1) # Output: True
```

### Que. 2

append():

- Adds a single element to the end of the list.
- Syntax: list.append(element)

example -

```
list1 = [1, 2, 3]
list1.append(4)
print(list1) # Output: [1, 2, 3, 4]
```

- Inserts an element at a specified position in the list.
- Syntax: list.insert(index, element)

example -

```
list1 = [1, 2, 4]
list1.insert(2, 3)
print(list1) # Output: [1, 2, 3, 4]
```

`remove()`:

- Removes the first occurrence of a specified value in the list.
- Syntax: `list.remove(element)`

example -

```
list1 = [1, 2, 3, 2]
list1.remove(2)
print(list1) # Output: [1, 3, 2]
```

`pop()`:

- Removes and returns the element at a specified index (or the last element if no index is provided).
- Syntax: `list.pop([index])`

example -

```
list1 = [1, 2, 3]
last_element = list1.pop()
print(last_element) # Output: 3
print(list1)       # Output: [1, 2]

specific_element = list1.pop(0)
print(specific_element) # Output: 1
print(list1)          # Output: [2]
```