List Methods

PRESENTED BY:-

KARAN V DAYAL

ASSITANT PROFESSOR(CSE DEPT.)

List Methos:-

List of Common Python List Methods

- `append()`
- `extend()`
- `insert()`
- `remove()`
- `pop()`
- `index()`
- `count()`
- `sort()`
- `reverse()`
- `copy()`

append() Method

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

```
my_list = [1, 2, 3]

my_list.append(4)

print(my_list)  # Output: [1, 2, 3, 4]
```

extend() Method

Adds all elements from another list to the end of the current list.

Syntax: `list.extend(iterable)`

Example:

```
my_list = [1, 2, 3]

my_list.extend([4, 5, 6])

print(my_list)  # Output: [1, 2, 3, 4, 5, 6]
```

`insert()` Method

- Inserts an element at a specific index.
- **Syntax**: `list.insert(index, element)`
- **Example**:

```
my_list = [1, 2, 3]

my_list.insert(1, 5)

print(my_list)  # Output: [1, 5, 2, 3]
```

`remove()` Method

- Removes the first occurrence of the specified element.
- **Syntax**: `list.remove(element)`
- **Example**:

```
my_list = [1, 2, 3, 2]

my_list.remove(2)

print(my_list)  # Output: [1, 3, 2]
```

`pop()` Method

- Removes and returns the element at the specified index (defaults to the last element).
- **Syntax**: `list.pop([index])`
- **Example**:

```
my_list = [1, 2, 3]

removed_item = my_list.pop()

print(my_list)          # Output: [1, 2]

print(removed_item)     # Output: 3
```

`index()` Method

- Returns the index of the first occurrence of the specified element.
- **Syntax**: `list.index(element)`
- **Example**:

```
my_list = [1, 2, 3]

idx = my_list.index(2)

print(idx)  # Output: 1
```

`count()` Method

- Returns the number of occurrences of a specific element.
- **Syntax**: `list.count(element)`
- **Example**:

```
my_list = [1, 2, 2, 3]

count_of_twos = my_list.count(2)

print(count_of_twos)  # Output: 2
```

`sort()` Method

- sort() Method
- **Content**:
  - Sorts the list in ascending order (can take parameters for reverse order).
  - **Syntax**: `list.sort([reverse=False])`
  - **Example**:

```
my_list = [3, 1, 2]

my_list.sort()

print(my_list)  # Output: [1, 2, 3]
```

`reverse()` Method

- Reverses the elements of the list in place.
- **Syntax**: `list.reverse()`
- **Example**:

```
my_list = [1, 2, 3]

my_list.reverse()

print(my_list)  # Output: [3, 2, 1]
```

`copy()` Method

- Returns a shallow copy of the list.
- **Syntax**: `list.copy()`
- **Example**:

```
my_list = [1, 2, 3]

copy_list = my_list.copy()

print(copy_list)  # Output: [1, 2, 3]
```

clear() Method

Syntax:

list.clear()

Description:

- Removes **all elements** from the list.

Example:

my_list = [1, 2, 3]my_list.clear()print(my_list) # Output: []

Summary of list methods:-

- **Modifying** lists: append(), extend(), insert(), remove(), pop()
- **Information retrieval**: index(), count()
- **Sorting & reversing**: sort(), reverse()
- **Miscellaneous**: clear(), copy()