

Practical No. 8 : Write python program to use built-in functions/methods on list: cmp, len, max, list, append, count, extend, insert, pop, remove, etc.

- **Practical related questions**

1. **Write syntax for a method to reverse a list.**



The `reverse()` method is used to reverse a list. It reverses the order of elements in the list in-place, meaning the original list is modified.

Syntax -

`list_name.reverse()`

Example -

```
my_list = [1, 2, 3, 4, 5]
print("Original list:", my_list)
my_list.reverse()
print("Reversed list:", my_list)
```

Output -

```
Original list: [1, 2, 3, 4, 5]
Reversed list: [5, 4, 3, 2, 1]
```

2. **Describe various list functions.**



- a) `append()` -

Adds an element to the end of the list.

Syntax -

`list.append(element)`

b) insert() -

Inserts an element at a specified index in the list. Shifts the subsequent elements to the right.

Syntax -

```
list.insert(index, element)
```

c) remove() -

Removes the first occurrence of the specified element from the list.

Syntax -

```
list.remove(element)
```

d) pop() -

Removes and returns the element at the specified index. If no index is specified, it removes and returns the last element. Raises an **IndexError** if the list is empty.

Syntax -

- 1) `list.pop(index)`
- 2) `list.pop()`

e) clear() -

Removes all elements from the list, leaving it empty.

Syntax -

```
list.clear()
```

f) sort() -

Sorts the list in ascending order by default. You can pass the **reverse=True** argument to sort in descending order.

Syntax -

- 1) `list.sort()`
- 2) `list.sort(reverse=True)`

g) `reverse()` -

Reverses the order of elements in the list.

Syntax -

```
list.reverse()
```

3. Describe the use of pop operator in list.



The `pop()` method in Python is used to remove and return an element from a list. This method can either remove the element at a specified index or remove the last element if no index is provided.

Syntax -

```
list_name.pop(index)
```

Example -

```
my_list = [10, 20, 30, 40]
removed_item = my_list.pop(1)
print(my_list)
print(removed_item)
```

Output -

```
[10, 30, 40]
20
```

4. Describe the use extend & pop method in list.



a) `extend()` Method:

The `extend()` method is used to add all elements from an iterable (like a list, tuple, or set) to the end of the current list. This method modifies the original list and does not return a new list.

Syntax:

```
list_name.extend(iterable)
```

Example:

```
list1 = [1, 2, 3]
list2 = [4, 5]
list1.extend(list2)
print(list1)
```

Output:

[1, 2, 3, 4, 5]

b) pop() Method:

The `pop()` method is used to remove and return an element at a specific index in a list. If no index is specified, it removes and returns the last element. It modifies the original list by removing the element.

Syntax :

```
list_name.pop(index)
list_name.pop()
```

Example:

```
my_list = [10, 20, 30, 40]
removed_item = my_list.pop(1)
print(my_list)
print(removed_item)
removed_item = my_list.pop()
print(my_list)
print(removed_item)
```

Output:

[10, 30, 40]
20
[10, 30]
40

5. Write a Python program to find common items from two lists.

→

```
list1 = [10, 20, 30, 40, 50]
list2 = [30, 40, 50, 60, 70]
common_items = [item for item in list1 if item in list2]
```

```
print("Common items:", common_items)
```

Output:

Common items: [30, 40, 50]

6. Write a Python program to sort a list.

```
my_list = [30, 10, 50, 20, 40]  
my_list.sort()  
print("Sorted list:", my_list)
```

Output:

Sorted list: [10, 20, 30, 40, 50]

