Practical No 4(a)

Aim: - Implementing Lists in Python.

Theory: -

List-

Lists are used to store multiple items in a single variable.

Lists are created using square brackets:

Create a List:

thislist = ["apple", "banana", "cherry"]

print(thislist)

List Items

List items are ordered, changeable, and allow duplicate values.

List items are indexed, the first item has index [0], the second item has index [1] etc.

Ordered

When it says that lists are ordered, it means that the items have a defined order, and that order will not change. If you add new items to a list, the new items will be placed at the end of the list.

Changeable

The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.

Allow Duplicates

Since lists are indexed, lists can have items with the same value:

Lists allow duplicate values:

```
thislist = ["apple", "banana", "cherry", "apple", "cherry"]
print(thislist)
```

List Length

To determine how many items a list has, use the len() function:

```
thislist = ["apple", "banana", "cherry"]
print(len(thislist))
```

List Items - Data Types

List items can be of any data type:

```
list1 = ["apple", "banana", "cherry"]
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]
```

A list with strings, integers and boolean values:

```
list1 = ["abc", 34, True, 40, "male"]
```

Access Items

List items are indexed and you can access them by referring to the index number:

Print the second item of the list:

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

Output:

```
['banana’]
```

Range of Indexes

You can specify a range of indexes by specifying where to start and where to end the range.

When specifying a range, the return value will be a new list with the specified items.

Return the third, fourth, and fifth item:

```
thelist= ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thelist[2:5])
```

Output:

```
['cherry', 'orange', 'kiwi']
```

Check if Item Exists

To determine if a specified item is present in a list use the in keyword:

```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
    print("Yes, 'apple' is in the fruits list")
```

Output:

```
Yes, 'apple' is in the fruits list
```

Change Item Value

To change the value of a specific item, refer to the index number:

Change the second item:

```
thislist = ["apple", "banana", "cherry"]
thislist[1] = "blackcurrant"
print(thislist)
```

Output:

```
['apple', 'blackcurrant', 'cherry']
```

Append Items
To add an item to the end of the list, use the append() method:

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```
Output:
```
['apple', 'banana', 'cherry', 'orange']
```

Insert Items
To insert a list item at a specified index, use the insert() method.
The **insert()** method inserts an item at the specified index:

Insert an item as the second position:

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(1, "orange")
print(thislist)
```
Output:
```
['apple', 'orange', 'banana', 'cherry']
```

Remove Specified Item
The remove() method removes the specified item.
```
thislist = ["apple", "banana", "cherry"]
thislist.remove("banana")
print(thislist)
```

Output:
```
['apple','cherry']
```

Sort List Alphanumerically
List objects have a sort() method that will sort the list alphanumerically, ascending, by default:
```
thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]
thislist.sort()
print(thislist)
```
Output:
```
['banana', 'kiwi', 'mango', 'orange', 'pineapple']
```

Sort the list numerically:
```
thislist = [100, 50, 65, 82, 23]
thislist.sort()
print(thislist)
```

Output:

```
[23, 50, 65, 82, 100]
```

Copy a List

You cannot copy a list simply by typing list2 = list1, because: list2 will only be a *reference* to list1, and changes made in list1 will automatically also be made in list2.
Make a copy of a list with the **copy()** method:

```
thislist = ["apple", "banana", "cherry"]
mylist = thislist.copy()
print(mylist)
```

Output:

```
['apple', 'banana', 'cherry']
```

Join Two Lists

There are several ways to join, or concatenate, two or more lists in Python.
One of the easiest ways are by using the + operator.

```
list1 = ["a", "b", "c"]
list2 = [1, 2, 3]

list3 = list1 + list2
print(list3)
```

Output:

```
['a', 'b', 'c', 1, 2, 3]
```

Results: So we studied the implementation of list and operations on list using Python.