Practical No. 4

Date:

Aim: Implementing Lists in python.

Practical No. 4

Date :

Aim : Implementing Lists in python

Theory : List are used to store multiple items in a single variable.
List 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 definite order, & 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 Duplicate Values.
Since lists are indexed, lists can have items with the same value.
code:

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

List length:
To determine how many items a list has, use a len() function.

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

List Items - Data types
List items can be any data types:

```
list_1 = ["apple", "banana", "cherry"]
list_2 = [1, 5, 7, 9, 3]
list_3 = [True, False, False, True]
```

A list with strings, integers & boolean values:

```
list_1 = ["abc", 34, True, 40, "bcd", "hex"]
```

Access items:

list items are indexed & you can access them by refering 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 specify -ing where to start and end the range . When specifying a range, the return value will be a new list with the specified items.

Return the 3rd, 4th & 5th item :

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

Output → ['cherry', 'orange', 'kiwi']

Check if item exsists :

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

```
thisapple = ["apple", "banana", "cherry"]
  if "cherry" in thisapple :
      print ("Yes, it is in fruits)
```

Output → Yes, it is in fruits.

Change item Value :

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

Change the second number :

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

Output → ['apple', 'Peach', 'Cherry']

Append items :

To add an item to the end of the list, use the append() method.

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

Output → ['apple', 'banana', 'cherry', 'orange', 'Peach']

Insert items :

To insert a list items at a specified index, use the insert () method

The insert() method inserts an item at the specified index :

Insert an item as the 2nd 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, 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]

Result: So, we Studied the implementation of the lists & operators on lists using python.

Copy a list :

You cannot copy a list simply by typing list 2 = list 1, because : list 2 will be only a ref. to list 1, & changes made in list 1 will automatically also be made in list 2.

Make a copy of a list with the copy () method :

```
thismethodlist = ["apple", "banana", "cherry"]
mylist = thismethodlist.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]
list 3 = list1 + list 2
print (list3)
```

Output → ['a', 'b', 'c', '1', '2', '3']

Result : So, we studied the implementation of lists & operations on lists using python.