LIST

1. What is a list ?

A list is a container that holds many objects under a single name.

2. How to define a list ?

Syntax to define a list is as follows:

listName = [object1, object2, object3]

Example: `bestFriends = ['Mark', 'Mary', 'Maria', 'John’]`

3. How to access the values in a list ?

The values in a list can be accessed by specifying the index of that value. Values are indexed starting from zero. From the bestFriends list, values can be accessed as mentioned below:

`bestFriends[0]` returns `‘Mark’`

`bestFriends[1]` returns `‘Mary’`

`bestFriends[2]` returns `‘Maria’`

`bestFriends[3]` returns `‘John’`

`bestFriends[4]` returns an index out of range error as our list contains only four elements

4. What are some of the list operations ?

1. Append

Append operation is used to add a new element to end of the list.

Syntax: list.append(element)

Example: `bestFriends.append(“Febin”)`

bestFriends list now returns the list with the name Febin appended to the end of the list

```
['Mark', 'Mary', 'Maria', 'John’, 'Febin']
```

2. Insert

Insert operation is used to add a new element at a specified index and shift the other elements to the right.

Syntax: list.insert(index, element)

Example: `bestFriends.insert(1, “Ben”)`

bestFriends list now returns the previous list along with name Ben placed at index 1

```
['Mark', 'Ben', 'Mary', 'Maria', 'John’, 'Febin']
```

3. Remove

Remove operation is used to remove an element from the list

Syntax: list.remove(element)

Example: `bestFriends.remove(“Mary”)`

bestFriends list now returns the list without the name Mary

```
['Mark', 'Ben', 'Maria', 'John’, 'Febin']
```

4. Sort

Sort operation is used to sort a given list in ascending order.

Syntax: list.sort()

Example: `bestFriends.sort()`

bestFriends list now gets sorted alphabetically as

```
['Ben', 'Febin', 'John’, 'Maria', 'Mark']
```

5. Reverse

Reverse operation is used to reverse a list.

Syntax: list.reverse()

Example: `bestFriends.reverse()`

bestFriends list now returns the list in reverse order with the name Ben now being at the end of the list

```
['Mark', 'Maria', 'John’, 'Febin', 'Ben']
```

6. Pop

Pop operation is used to return an element at the specified index and remove it from the list.

Syntax: list.pop(index)

Example: `bestFriends.pop(2)` returns `‘John’`

bestFriends list now the list without the name John in it

```
['Mark', 'Maria', 'Febin', 'Ben']
```

If no index is specified, pop operation returns the last element of the list.

Example: `bestFriends.pop()` returns `‘Ben’`

bestFriends list now returns the list with the name Ben removed from the end of the list

```
['Mark', 'Maria', 'Febin']
```