

JAVA - THE LIST INTERFACE

The List interface extends **Collection** and declares the behavior of a collection that stores a sequence of elements.

- Elements can be inserted or accessed by their position in the list, using a zero-based index.
- A list may contain duplicate elements.
- In addition to the methods defined by **Collection**, List defines some of its own, which are summarized in the following below Table.
- Several of the list methods will throw an `UnsupportedOperationException` if the collection cannot be modified, and a `ClassCastException` is generated when one object is incompatible with another.

SN	Methods with Description
1	void add(int index, Object obj) Inserts obj into the invoking list at the index passed in index. Any pre-existing elements at or beyond the point of insertion are shifted up. Thus, no elements are overwritten.
2	boolean addAll(int index, Collection c) Inserts all elements of c into the invoking list at the index passed in index. Any pre-existing elements at or beyond the point of insertion are shifted up. Thus, no elements are overwritten. Returns true if the invoking list changes and returns false otherwise.
3	Object get(int index) Returns the object stored at the specified index within the invoking collection.
4	int indexOf(Object obj) Returns the index of the first instance of obj in the invoking list. If obj is not an element of the list, -1 is returned.
5	int lastIndexOf(Object obj) Returns the index of the last instance of obj in the invoking list. If obj is not an element of the list, -1 is returned.
6	ListIterator listIterator() Returns an iterator to the start of the invoking list.
7	ListIterator listIterator(int index) Returns an iterator to the invoking list that begins at the specified index.
8	Object remove(int index) Removes the element at position index from the invoking list and returns the deleted element. The resulting list is compacted. That is, the indexes of subsequent elements are decremented by one

9 Object set(int index, Object obj)

Assigns obj to the location specified by index within the invoking list.

10 List subList(int start, int end)

Returns a list that includes elements from start to end-1 in the invoking list. Elements in the returned list are also referenced by the invoking object.

Example:

Above interface has been implemented in various classes like ArrayList or LinkedList, etc. Following is the example to explain few methods from various class implementation of the above collection methods:

```
import java.util.*;

public class CollectionsDemo {

    public static void main(String[] args) {
        List a1 = new ArrayList();
        a1.add("Zara");
        a1.add("Mahnaz");
        a1.add("Ayan");
        System.out.println(" ArrayList Elements");
        System.out.print("\t" + a1);

        List l1 = new LinkedList();
        l1.add("Zara");
        l1.add("Mahnaz");
        l1.add("Ayan");
        System.out.println();
        System.out.println(" LinkedList Elements");
        System.out.print("\t" + l1);
    }
}
```

This would produce the following result:

```
ArrayList Elements
    [Zara, Mahnaz, Ayan]
LinkedList Elements
    [Zara, Mahnaz, Ayan]
```