

ArrayList vs LinkedList in Java

Two popular lists in Java are:

1. **ArrayList**:-Implemented with the concept of dynamic array.

```
ArrayList<Type> arrL = new ArrayList<Type>();
```

Here Type is the data type of elements in ArrayList to be created

2. **LinkedList**:-Implemented with the concept of doubly linked list.

```
LinkedList<Type> linkL = new LinkedList<Type>();
```

Here Type is the data type of elements in LinkedList to be created

Comparision between **ArrayList** and **LinkedList**:-

1. Insertions are easy and fast in LinkedList as compared to ArrayList because there is no risk of resizing array and copying content to new array if array gets full which makes adding into ArrayList of $O(n)$ in worst case, while adding is $O(1)$ operation in LinkedList in Java. ArrayList also needs to be update its index if you insert something anywhere except at the end of array.
2. Removal also better in LinkedList than ArrayList due to same reasons as insertion.
3. LinkedList has more memory overhead than ArrayList because in ArrayList each index only holds actual object (data) but in case of LinkedList each node holds both data and address of next and previous node.
4. Both LinkedList and ArrayList require $O(n)$ time to find if an element is present or not. However we can do Binary Search on ArrayList if it is sorted and therefore can search in $O(\log n)$ time.

// Java program to demonstrate difference between ArrayList and

```
// LinkedList.
import java.util.ArrayList;
import java.util.LinkedList;

public class ArrayListLinkedListExample
{
    public static void main(String[] args)
    {
        ArrayList<String> arrlistobj = new ArrayList<String>();
        arrlistobj.add("0. Practice.GeeksforGeeks.org");
        arrlistobj.add("1. Quiz.GeeksforGeeks.org");
        arrlistobj.add("2. Code.GeeksforGeeks.org");
        arrlistobj.remove(1); // Remove value at index 2
        System.out.println("ArrayList object output : " + arrlistobj);

        // Checking if an element is present.
        if (arrlistobj.contains("2. Code.GeeksforGeeks.org"))
            System.out.println("Found");
        else
            System.out.println("Not found");

        LinkedList llobj = new LinkedList();
        llobj.add("0. Practice.GeeksforGeeks.org");
        llobj.add("1. Quiz.GeeksforGeeks.org");
        llobj.add("2. Code.GeeksforGeeks.org");
        llobj.remove("1. Quiz.GeeksforGeeks.org");
        System.out.println("LinkedList object output : " + llobj);

        // Checking if an element is present.
        if (llobj.contains("2. Code.GeeksforGeeks.org"))
            System.out.println("Found");
        else
            System.out.println("Not found");
    }
}
```

[Run on IDE](#)

Output:

```
ArrayList object output :[0. Practice.GeeksforGeeks.org, 2. Code.GeeksforGeeks.org]
Found
LinkedList object output :[0. Practice.GeeksforGeeks.org, 2. Code.GeeksforGeeks.org]
Found
```

This article is contributed by **Pratik Agarwal**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

Java Java-Collections