

## Java LinkedList

### **Java LinkedList**

In the previous chapter, you learned about the [ArrayList](#) class. The [LinkedList](#) class is almost identical to the [ArrayList](#):

#### **Example**

*// Import the LinkedList class*

---

```
import java.util.LinkedList;
```

```
public class Main {  
    public static void main(String[] args) {  
        LinkedList<String> cars = new LinkedList<String>();  
        cars.add("Volvo");  
        cars.add("BMW");  
        cars.add("Ford");  
        cars.add("Mazda");  
        System.out.println(cars);  
    }  
}
```

---

## ArrayList vs. LinkedList

The `LinkedList` class is a collection which can contain many objects of the same type, just like the `ArrayList`.

The `LinkedList` class has all of the same methods as the `ArrayList` class because they both implement the `List` interface. This means that you can add items, change items, remove items and clear the list in the same way.

However, while the `ArrayList` class and the `LinkedList` class can be used in the same way, they are built very differently.

### How the ArrayList works

The `ArrayList` class has a regular array inside it. When an element is added, it is placed into the array. If the array is not big enough, a new, larger array is created to replace the old one and the old one is removed.

### How the LinkedList works

The `LinkedList` stores its items in "containers." The list has a link to the first container and each container has a link to the next container in the list. To add an element to the list, the element is placed into a new container and that container is linked to one of the other containers in the list.

### When To Use

Use an `ArrayList` for storing and accessing data, and `LinkedList` to manipulate data.

### LinkedList Methods

For many cases, the `ArrayList` is more efficient as it is common to need access to random items in the list, but the `LinkedList` provides several methods to do certain operations more efficiently:

**addFirst()****Adds an item to the beginning of the list.**

```
import java.util.LinkedList;

public class Main {
    public static void main(String[] args) {
        LinkedList<String> cars = new LinkedList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");

        // Use addFirst() to add the item to the beginning
        cars.addFirst("Mazda");
        System.out.println(cars);
    }
}
```

**addLast()****Add an item to the end of the list**

```
import java.util.LinkedList;

public class Main {
    public static void main(String[] args) {
        LinkedList<String> cars = new LinkedList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");

        // Use addLast() to add the item to the end
        cars.addLast("Mazda");
        System.out.println(cars);
    }
}
```

**removeFirst()****Remove an item from the beginning of the list.**

```
import java.util.LinkedList;

public class Main {
    public static void main(String[] args) {
        LinkedList<String> cars = new LinkedList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");

        // Use removeFirst() remove the first item from the list
        cars.removeFirst();
        System.out.println(cars);
    }
}
```

**removeLast()****Remove an item from the end of the list**

```
import java.util.LinkedList;

public class Main {
    public static void main(String[] args) {
        LinkedList<String> cars = new LinkedList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");

        // Use removeLast() remove the last item from the list
        cars.removeLast();
        System.out.println(cars);
    }
}
```

**getFirst()****Get the item at the beginning of the list**

```
import java.util.LinkedList;

public class Main {
    public static void main(String[] args) {
        LinkedList<String> cars = new LinkedList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");

        // Use getFirst() to display the first item in the list
        System.out.println(cars.getFirst());
    }
}
```

**getLast()****Get the item at the end of the list**

```
import java.util.LinkedList;

public class Main {
    public static void main(String[] args) {
        LinkedList<String> cars = new LinkedList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");

        // Use getLast() to display the last item in the list
        System.out.println(cars.getLast());
    }
}
```