**How to loop ArrayList in Java**

[**JAVA COLLECTIONS**](http://beginnersbook.com/category/java-collections/)

Earlier we shared **[ArrayList example](http://beginnersbook.com/2013/12/java-arraylist/" \t "_blank)** and [**how to initialize ArrayList in Java**](http://beginnersbook.com/2013/12/how-to-initialize-an-arraylist/). In this post we are sharing how to**iterate (loop) ArrayList in Java**.

There are four ways to loop ArrayList:S

1. For Loop
2. Advanced for loop
3. While Loop
4. Iterator

Lets have a look at the below example – I have used all of the mentioned methods for iterating list.

import java.util.\*;

public class LoopExample {

public static void main(String[] args) {

ArrayList<Integer> arrlist = new ArrayList<Integer>();

arrlist.add(14);

arrlist.add(7);

arrlist.add(39);

arrlist.add(40);

/\* For Loop for iterating ArrayList \*/

System.out.println("For Loop");

for (int counter = 0; counter < arrlist.size(); counter++) {

System.out.println(arrlist.get(counter));

}

/\* Advanced For Loop\*/

System.out.println("Advanced For Loop");

for (Integer num : arrlist) {

System.out.println(num);

}

/\* While Loop for iterating ArrayList\*/

System.out.println("While Loop");

int count = 0;

while (arrlist.size() > count) {

System.out.println(arrlist.get(count));

count++;

}

/\*Looping Array List using Iterator\*/

System.out.println("Iterator");

Iterator iter = arrlist.iterator();

while (iter.hasNext()) {

System.out.println(iter.next());

}

}

}

Output:

For Loop

14

7

39

40

Advanced For Loop

14

7

39

40

While Loop

14

7

39

40

Iterator

14

7

39

40

In the comment section below, Govardhan asked a question: He asked, how to iterate an ArrayList using [**Enumeration**](https://docs.oracle.com/javase/7/docs/api/java/util/Enumeration.html). Govardhan here is the code:

**How to iterate arraylist elements using Enumeration interface**

import java.util.Enumeration;

import java.util.ArrayList;

import java.util.Collections;

public class EnumExample {

public static void main(String[] args) {

//create an ArrayList object

ArrayList<String> arrayList = new ArrayList<String>();

//Add elements to ArrayList

arrayList.add("C");

arrayList.add("C++");

arrayList.add("Java");

arrayList.add("DotNet");

arrayList.add("Perl");

// Get the Enumeration object

Enumeration<String> e = Collections.enumeration(arrayList);

// Enumerate through the ArrayList elements

System.out.println("ArrayList elements: ");

while(e.hasMoreElements())

System.out.println(e.nextElement());

}

}

Output:

ArrayList elements:

C

C++

Java

DotNet

Perl

public interface **Enumeration<E>**

An object that implements the Enumeration interface generates a series of elements, one at a time. Successive calls to the nextElement method return successive elements of the series.

For example, to print all elements of a Vector<E> *v*:

for (Enumeration<E> e = v.elements(); e.hasMoreElements();)

System.out.println(e.nextElement());

Methods are provided to enumerate through the elements of a vector, the keys of a hashtable, and the values in a hashtable. Enumerations are also used to specify the input streams to a SequenceInputStream.