

JAVA.UTIL.ARRAYLIST.TOARRAYT[] METHOD

http://www.tutorialspoint.com/java/util/arraylist_toarray.htm

Copyright © tutorialspoint.com

Description

The **java.util.ArrayList.toArrayT[]** method returns an array containing all of the elements in this list in proper sequence *from first to last element*. Following are the important points about ArrayList.toArray:

- The runtime type of the returned array is that of the specified array. If the list fits in the specified array, it is returned therein. Otherwise, a new array is allocated with the runtime type of the specified array and the size of this list.
- If the list fits in the specified array with room to spare *i. e. , the array has more elements than the list*, the element in the array immediately following the end of the collection is set to null.

Declaration

Following is the declaration for **java.util.ArrayList.toArray** method

```
public <T> T[] toArray(T[] a)
```

Parameters

- **a** -- This is the array into which the elements of the list are to be stored, if it is big enough; otherwise, a new array of the same runtime type is allocated for this purpose.

Return Value

This method returns an array containing the elements of the list.

Exception

- **ArrayStoreException** -- If the runtime type of the specified array is not a supertype of the runtime type of every element in this list.
- **NullPointerException** -- If the specified array is null.

Example

The following example shows the usage of java.util.ArrayList.toArray method.

```
package com.tutorialspoint;

import java.util.ArrayList;

public class ArrayListDemo {
    public static void main(String[] args) {

        // create an empty array list with an initial capacity
        ArrayList<Integer> arrlist = new ArrayList<Integer>();

        // use add() method to add values in the list
        arrlist.add(10);
        arrlist.add(12);
        arrlist.add(31);
        arrlist.add(49);

        System.out.println("Printing elements of array1");

        // let us print all the elements available in list
        for (Integer number : arrlist) {
            System.out.println("Number = " + number);
        }
    }
}
```

```

}

// toArray copies content into other array
Integer list2[] = new Integer[arrlist.size()];
list2 = arrlist.toArray(list2);

System.out.println("Printing elements of array2");

// let us print all the elements available in list
for (Integer number : list2) {
    System.out.println("Number = " + number);
}
}
}

```

Let us compile and run the above program, this will produce the following result:

```

Printing elements of array1
Number = 10
Number = 12
Number = 31
Number = 49
Printing elements of array2
Number = 10
Number = 12
Number = 31
Number = 49

```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js