JAVA.UTIL.VECTOR.TOARRAY METHOD

Description

The **toArray*T*[]*a*** method is used to return an array containing all of the elements in this Vector in the correct order.

Declaration

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

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

Parameters

- **a** -- This is the array into which the elements of the Vector 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

The method call returns an array containing the elements of the Vector.

Exception

- **ArrayStoreException** -- This exception is thrown if the runtime type of a is not a supertype of the runtime type of every element in this Vector.
- **NullPointerException** -- This exception is thrown if if the given array is null.

Example

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

```
package com.tutorialspoint;

import java.util.*;
import java.util.Arrays;

public class VectorDemo {
   public static void main(String[] args) {
   // create an empty Vector vec with an initial capacity of 4
   Vector<Integer> vec = new Vector<Integer>(4);
   // create an array
   Integer[] anArray = new Integer[4];

   // use add() method to add elements in the vector
   vec.add(4);
   vec.add(3);
   vec.add(2);
   vec.add(1);

   // fill the array from the vector
   vec.toArray(anArray);

   // check the content of the array
   System.out.println("Elements are: ");
   for (int i = 0; i < anArray.length; i++) {
   System.out.println(anArray[i]);
   }
   }
}
```

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

```
Elements are:
4
3
2
1
```