

# JAVA.UTIL.VECTOR.ADD METHOD

[http://www.tutorialspoint.com/java/util/vector\\_add\\_index.htm](http://www.tutorialspoint.com/java/util/vector_add_index.htm)

Copyright © tutorialspoint.com

## Description

This **add(int index, Element)** method is used to Insert the specified element *E* at the specified position *index* in the list. While inserting the **new element**, it Shifts the **existing element** currently at that position *if any* and any subsequent elements to the right with changed **indices**. This is another variation of the previous add method.

## Declaration

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

```
public void add(int index, E element)
```

## Parameters

- **index** -- This is the index *position* at which the specified element is to be inserted.
- **element** -- This is the element to be inserted at the specified location.

## Return Value

The return type is **void** so the method does not return any value.

## Exception

- **IndexOutOfBoundsException** The method throws this exception if the index *element position* we are trying to access is out of range  $index < 0 \mid index > size()$ .

## Example

The following example shows the usage of `java.util.Vector.add(int index, Element)` method.

```
package com.tutorialspoint;

import java.util.Vector;

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);

        // use add() method to add elements in the vector
        vec.add(0,4);
        vec.add(1,3);
        vec.add(2,2);
        vec.add(3,1);

        // let us print all the elements available in vector
        System.out.println("Added numbers are :- ");
        for (Integer number : vec) {
            System.out.println("Index :"+vec.indexOf(number) +" Number: " + number);
        }

        // added new number 10 at 3rd position/index
        vec.add(3,10);

        // let us print all the elements available in vector after insertion
        System.out.println("Added numbers after insertion are :- ");
        for (Integer number : vec) {
            System.out.println("Index :"+vec.indexOf(number) +" Number: " + number);
        }
    }
}
```

```
}  
}  
}
```

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

```
Added numbers are :-  
Index :0 Number: 4  
Index :1 Number: 3  
Index :2 Number: 2  
Index :3 Number: 1  
Added numbers after insertion are :-  
Index :0 Number: 4  
Index :1 Number: 3  
Index :2 Number: 2  
Index :3 Number: 10  
Index :4 Number: 1
```

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