

# JAVA.UTIL.ARRAYDEQUE.PEEK METHOD

[http://www.tutorialspoint.com/java/util/arraydeque\\_peek.htm](http://www.tutorialspoint.com/java/util/arraydeque_peek.htm)

Copyright © tutorialspoint.com

## Description

The **java.util.ArrayDeque.peek** method retrieves the **head** of the queue *but does not remove* represented by this **deque**. Returns **null** if this deque is empty.

## Declaration

Following is the declaration for **java.util.ArrayDeque.peek** method

```
public E peek()
```

## Parameters

- NA

## Return Value

This method returns the **head** of the queue represented by this deque, or **null** if this deque is empty.

## Exception

- NA

## Example

The following example shows the usage of java.util.ArrayDeque.peek method.

```
package com.tutorialspoint;

import java.util.ArrayDeque;
import java.util.Deque;

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

        // create an empty array deque with an initial capacity
        Deque<Integer> deque = new ArrayDeque<Integer>(8);

        // use add() method to add elements in the deque
        deque.add(25);
        deque.add(30);
        deque.add(20);
        deque.add(18);

        // this will retrieve head of the queue
        int retval = deque.peek();
        System.out.println("Retrieved Element is " + retval);

        // printing all the elements available in deque
        for (Integer number : deque) {
            System.out.println("Number = " + number);
        }
    }
}
```

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

```
Retrieved Element is 25
Number = 25
```

Number = 30  
Number = 20  
Number = 18

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