

JAVA.MATH.BIGINTEGER.TOBYTEARRAY METHOD

http://www.tutorialspoint.com/java/math/biginteger_tobytearray.htm

Copyright © tutorialspoint.com

Description

The **java.math.BigInteger.toByteArray** returns a byte array containing the two's-complement representation of this BigInteger. The byte array will be in big-endian byte-order: the most significant byte is in the zeroth element.

The array will contain the minimum number of bytes required to represent this BigInteger, including at least one sign bit, which is $\text{ceil}((\text{this.bitLength}() + 1)/8)$. This representation is compatible with the `byte[]` constructor.

Declaration

Following is the declaration for **java.math.BigInteger.toByteArray** method

```
public byte[] toByteArray()
```

Parameters

- NA

Return Value

This method returns a byte array containing the two's-complement representation of this BigInteger.

Exception

- NA

Example

The following example shows the usage of `math.BigInteger.toByteArray` method

```
package com.tutorialspoint;

import java.math.*;

public class BigIntegerDemo {

    public static void main(String[] args) {

        // create 2 BigInteger objects
        BigInteger bi1, bi2;

        // create 2 byte arrays
        byte b1[], b2[];

        // create and assign value to byte array b3
        byte b3[] = { 0x1, 0x00, 0x00 };

        bi1 = new BigInteger("10");
        bi2 = new BigInteger(b3); // using byte[] constructor of BigInteger

        // assign byte array representation of bi1, bi2 to b1, b2
        b1 = bi1.toByteArray();
        b2 = bi2.toByteArray();

        String str1 = "Byte array representation of " + bi1 + " is: ";

        System.out.println( str1 );
    }
}
```

```
// print byte array b1 using for loop
for (int i=0; i < b1.length; i++)
{
    System.out.format("0x%02X\n", b1[i]);
}

String str2 = "Byte array representation of " + b1 + " is: ";

System.out.println( str2 );

// print byte array b2 using for loop
for (int j=0; j < b2.length; j++)
{
    System.out.format("0x%02X ", b2[j]);
}
}
```

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

```
Byte array representation of 10 is:
0x0A
Byte array representation of 65536 is:
0x01 0x00 0x00
```

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