JAVA.MATH.BIGINTEGER.TOSTRING METHOD

Description

The **java.math.BigInteger.toString*intradix*** returns the String representation of this BigInteger in the given radix. If the radix is outside the range from Character.MIN_RADIX to Character.MAX_RADIX inclusive, it will default to 10 *asisthecaseforInteger. toString*.

The digit-to-character mapping provided by Character.forDigit is used, and a minus sign is prepended if appropriate. This representation is compatible with the *String, int* constructor.

Declaration

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

```
public String toString(int radix)
```

Parameters

- **radix** - radix of the String representation

Return Value

This method returns the String representation of this BigInteger in the given radix.

Exception

- **NA**

Example

The following example shows the usage of math.BigInteger.toString 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 String objects
 String s1, s2;

 bi1 = new BigInteger("16");
 bi2 = new BigInteger("-16");

 // assign String value of bi1, bi2 to s1, s2 using radix
 s1 = bi1.toString(8); // octal value of bi1
 s2 = bi2.toString(2); // binary value of bi2

 String str1 = "String value of " +bi1+ " in radix 8 is " +s1;
 String str2 = "String value of " +bi2+ " in radix 2 is " +s2;

 // print s1, s2 values
 System.out.println( str1 );
 System.out.println( str2 );
   }
}
```

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

```
String value of 16 in radix 8 is 20
String value of -16 in radix 2 is -10000
```