

JAVA.MATH.BIGDECIMAL.TOPLAINSTRING METHOD

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

Copyright © tutorialspoint.com

Description

The **java.math.BigDecimal.toPlainString** returns a string representation of this BigDecimal without an exponent field. For values with a positive scale, the number of digits to the right of the decimal point is used to indicate scale.

For values with a zero or negative scale, the resulting string is generated as if the value were converted to a numerically equal value with zero scale and as if all the trailing zeros of the zero scale value were present in the result.

The entire string is prefixed by a minus sign character '-' '\u002D' if the unscaled value is less than zero. No sign character is prefixed if the unscaled value is zero or positive.

If the result of this method is passed to the string constructor, only the numerical value of this BigDecimal will necessarily be recovered; the representation of the new BigDecimal may have a different scale.

In particular, if this BigDecimal has a negative scale, the string resulting from this method will have a scale of zero when processed by the string constructor.

Declaration

Following is the declaration for **java.math.BigDecimal.toPlainString** method

```
public String toPlainString()
```

Parameters

- NA

Return Value

This method returns a string representation of this BigDecimal without an exponent field.

Exception

- NA

Example

The following example shows the usage of math.BigDecimal.toPlainString method

```
package com.tutorialspoint;

import java.math.*;

public class BigDecimalDemo {

    public static void main(String[] args) {

        // create a BigDecimal object
        BigDecimal bg;

        // create a String object
        String s;

        MathContext mc = new MathContext(3); // 3 precision

        bg = new BigDecimal("1234E+4", mc);
```

```
// assign the plain string value of bg to s
s = bg.toPlainString();

String str = "Plain string value of " + bg + " is " + s;

// print s value
System.out.println( str );
}
```

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

```
Plain string value of 1.23E+7 is 12300000
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```