

JAVA.LANG.DOUBLE.COMPARETO METHOD

Description

The **java.lang.Double.compareTo** method compares two Double objects numerically. There are two ways in which comparisons performed by this method differ from those performed by the Java language numerical comparison operators `<`, `<=`, `==`, `>=` when applied to primitive double values:

- `Double.NaN` is considered by this method to be equal to itself and greater than all other double values *including* `Double.POSITIVE_INFINITY`.
- `0.0d` is considered by this method to be greater than `-0.0d`.

Declaration

Following is the declaration for **java.lang.Double.compareTo** method

```
public int compareTo(Double anotherDouble)
```

Parameters

- **anotherDouble** -- This is the Double to be compared.

Return Value

This method returns the value `0` if `anotherDouble` is numerically equal to this `Double`; a value less than `0` if this `Double` is numerically less than `anotherDouble`; and a value greater than `0` if this `Double` is numerically greater than `anotherDouble`.

Exception

- **NA**

Example

The following example shows the usage of `java.lang.Double.compareTo` method.

```
package com.tutorialspoint;

import java.lang.*;

public class DoubleDemo {

    public static void main(String[] args) {

        // compares two Double objects numerically
        Double obj1 = new Double("8.5");
        Double obj2 = new Double("11.50");
        int retval = obj1.compareTo(obj2);

        if(retval > 0) {
            System.out.println("obj1 is greater than obj2");
        }
        else if(retval < 0) {
            System.out.println("obj1 is less than obj2");
        }
        else {
            System.out.println("obj1 is equal to obj2");
        }
    }
}
```

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

```
obj1 is less than obj2
```

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