Date class in Java (With Examples)

Difficulty Level : Easy Last Updated : 02 Jan, 2019

The class Date represents a specific instant in time, with millisecond precision. The Date class of java.util package implements Serializable, Cloneable and Comparable interface. It provides constructors and methods to deal with date and time with java.

Constructors

- **Date()** : Creates date object representing current date and time.
- **Date(long milliseconds)** : Creates a date object for the given milliseconds since January 1, 1970, 00:00:00 GMT.
- **Date(int year, int month, int date)**
- **Date(int year, int month, int date, int hrs, int min)**
- **Date(int year, int month, int date, int hrs, int min, int sec)**
- **Date(String s)**
  Note : The last 4 constructors of the Date class are Deprecated.

```
// Java program to demonstrate constuctors of Date
import java.util.*;

public class Main
{
    public static void main(String[] args)
    {
        Date d1 = new Date();
        System.out.println("Current date is " + d1);
        Date d2 = new Date(2323223232L);
        System.out.println("Date represented is "+ d2 );
    }
}
```

Output:

```
Current date is Tue Jul 12 18:35:37 IST 2016
Date represented is Wed Jan 28 02:50:23 IST 1970
```

Important Methods

- **boolean after(Date date) :** Tests if current date is after the given date.
- **boolean before(Date date) :** Tests if current date is before the given date.
- **int compareTo(Date date) :** Compares current date with given date. Returns 0 if the argument Date is equal to the Date; a value less than 0 if the Date is before the Date argument; and a value greater than 0 if the Date is after the Date argument.

- **long getTime()** : Returns the number of milliseconds since January 1, 1970, 00:00:00 GMT represented by this Date object.
- **void setTime(long time)** : Changes the current date and time to given time.

```
// Program to demonstrate methods of Date class
import java.util.*;

public class Main
{
    public static void main(String[] args)
    {
        // Creating date
        Date d1 = new Date(2000, 11, 21);
        Date d2 = new Date();  // Current date
        Date d3 = new Date(2010, 1, 3);

        boolean a = d3.after(d1);
        System.out.println("Date d3 comes after " +
                           "date d2: " + a);

        boolean b = d3.before(d2);
        System.out.println("Date d3 comes before "+
                           "date d2: " + b);

        int c = d1.compareTo(d2);
        System.out.println(c);

        System.out.println("Miliseconds from Jan 1 "+
                "1970 to date d1 is " + d1.getTime());

        System.out.println("Before setting "+d2);
        d2.setTime(204587433443L);
        System.out.println("After setting "+d2);
    }
}
```

Output:

```
Date d3 comes after date d2: true
Date d3 comes before date d2: false
1
Miliseconds from Jan 1 1970 to date d1 is 60935500800000
Before setting Tue Jul 12 13:13:16 UTC 2016
After setting Fri Jun 25 21:50:33 UTC 1976
```

This article is contributed by **Rahul Agrawal** .If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.