

# JSP - HANDLING DATE

[http://www.tutorialspoint.com/jsp/jsp\\_handling\\_date.htm](http://www.tutorialspoint.com/jsp/jsp_handling_date.htm)

Copyright © tutorialspoint.com

One of the most important advantages of using JSP is that you can use all the methods available in core Java. This tutorial would take you through Java provided **Date** class which is available in **java.util** package, this class encapsulates the current date and time.

The Date class supports two constructors. The first constructor initializes the object with the current date and time.

```
Date( )
```

The following constructor accepts one argument that equals the number of milliseconds that have elapsed since midnight, January 1, 1970

```
Date(long millisec)
```

Once you have a Date object available, you can call any of the following support methods to play with dates:

SN	Methods with Description
1	<b>boolean after</b> <i>Date date</i>  Returns true if the invoking Date object contains a date that is later than the one specified by date, otherwise, it returns false.
2	<b>boolean before</b> <i>Date date</i>  Returns true if the invoking Date object contains a date that is earlier than the one specified by date, otherwise, it returns false.
3	<b>Object clone</b>  Duplicates the invoking Date object.
4	<b>int compareTo</b> <i>Date date</i>  Compares the value of the invoking object with that of date. Returns 0 if the values are equal. Returns a negative value if the invoking object is earlier than date. Returns a positive value if the invoking object is later than date.
5	<b>int compareTo</b> <i>Object obj</i>  Operates identically to <i>compareToDate</i> if obj is of class Date. Otherwise, it throws a <i>ClassCastException</i> .
6	<b>boolean equals</b> <i>Object date</i>  Returns true if the invoking Date object contains the same time and date as the one specified by date, otherwise, it returns false.

- 7      **long getTime**
- Returns the number of milliseconds that have elapsed since January 1, 1970.
- 8      **int hashCode**
- Returns a hash code for the invoking object.
- 9      **void setTime***longtime*
- Sets the time and date as specified by time, which represents an elapsed time in milliseconds from midnight, January 1, 1970
- 10     **String toString**
- Converts the invoking Date object into a string and returns the result.

## Getting Current Date & Time

This is very easy to get current date and time in JSP program. You can use a simple Date object with *toString* method to print current date and time as follows:

```
<%@ page import="java.io.*,java.util.*, javax.servlet.*" %>
<html>
<head>
<title>Display Current Date & Time</title>
</head>
<body>
<center>
<h1>Display Current Date & Time</h1>
</center>
<%
    Date date = new Date();
    out.print( "<h2 align=\"center\">" +date.toString()+"</h2>");
%>
</body>
</html>
```

Now let us keep about code in CurrentDate.jsp and then call this JSP using URL <http://localhost:8080/CurrentDate.jsp>. This would produce following result:

### DISPLAY CURRENT DATE & TIME

---

**Mon Jun 21 21:46:49 GMT+04:00 2010**

Try to refresh URL <http://localhost:8080/CurrentDate.jsp> and you would find difference in seconds everytime you would refresh.

## Date Comparison:

As I mentioned above you can use all the available Java methods in your JSP scripts. In case you need to compare two dates, following are the methods:

- You can use *getTime* to obtain the number of milliseconds that have elapsed since midnight,

January 1, 1970, for both objects and then compare these two values.

- You can use the methods before, after, and equals. Because the 12th of the month comes before the 18th, for example, `new Date(99, 2, 12).before(new Date(99, 2, 18))` returns true.
- You can use the `compareTo` method, which is defined by the `Comparable` interface and implemented by `Date`.

## Date Formatting using SimpleDateFormat:

`SimpleDateFormat` is a concrete class for formatting and parsing dates in a locale-sensitive manner. `SimpleDateFormat` allows you to start by choosing any user-defined patterns for date-time formatting.

Let us modify above example as follows:

```
<%@ page import="java.io.*, java.util.*" %>
<%@ page import="javax.servlet.*, java.text.*" %>
<html>
<head>
<title>Display Current Date & Time</title>
</head>
<body>
<center>
<h1>Display Current Date & Time</h1>
</center>
<%
    Date dNow = new Date( );
    SimpleDateFormat ft =
    new SimpleDateFormat ("E yyyy.MM.dd 'at' hh:mm:ss a zzz");
    out.print( "<h2 align=\"center\">" + ft.format(dNow) + "</h2>");
%>
</body>
</html>
```

Compile above servlet once again and then call this servlet using URL `http://localhost:8080/CurrentDate`. This would produce following result:

### DISPLAY CURRENT DATE & TIME

**Mon 2010.06.21 at 10:06:44 PM GMT+04:00**

## Simple DateFormat format codes:

To specify the time format use a time pattern string. In this pattern, all ASCII letters are reserved as pattern letters, which are defined as the following:

Character	Description	Example
G	Era designator	AD
y	Year in four digits	2001
M	Month in year	July or 07
d	Day in month	10
h	Hour in A.M./P.M. 1 12	12
H	Hour in day 0 23	22
m	Minute in hour	30

s	Second in minute	55
S	Millisecond	234
E	Day in week	Tuesday
D	Day in year	360
F	Day of week in month	2 <i>secondWed. inJuly</i>
w	Week in year	40
W	Week in month	1
a	A.M./P.M. marker	PM
k	Hour in day 1 24	24
K	Hour in A.M./P.M. 0 11	10
z	Time zone	Eastern Standard Time
'	Escape for text	Delimiter
"	Single quote	`

For a complete list of constant available methods to manipulate date, you can refer to standard [java documentation](#)

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