

JAVASCRIPT - THE DATE OBJECT

http://www.tutorialspoint.com/javascript/javascript_date_object.htm

Copyright © tutorialspoint.com

The Date object is a datatype built into the JavaScript language. Date objects are created with the **new Date** as shown below.

Once a Date object is created, a number of methods allow you to operate on it. Most methods simply allow you to get and set the year, month, day, hour, minute, second, and millisecond fields of the object, using either local time or UTC *universal, or GMT* time.

The ECMAScript standard requires the Date object to be able to represent any date and time, to millisecond precision, within 100 million days before or after 1/1/1970. This is a range of plus or minus 273,785 years, so JavaScript can represent date and time till the year 275755.

Syntax

You can use any of the following syntaxes to create a Date object using Date constructor.

```
new Date( )  
new Date(milliseconds)  
new Date(datestring)  
new Date(year, month, date[, hour, minute, second, millisecond ])
```

Note – Parameters in the brackets are always optional.

Here is a description of the parameters –

- **No Argument** – With no arguments, the Date constructor creates a Date object set to the current date and time.
- **milliseconds** – When one numeric argument is passed, it is taken as the internal numeric representation of the date in milliseconds, as returned by the getTime method. For example, passing the argument 5000 creates a date that represents five seconds past midnight on 1/1/70.
- **datestring** – When one string argument is passed, it is a string representation of a date, in the format accepted by the **Date.parse** method.
- **7 arguments** – To use the last form of the constructor shown above. Here is a description of each argument:
 - **year** – Integer value representing the year. For compatibility *inordertoavoidtheY2Kproblem*, you should always specify the year in full; use 1998, rather than 98.
 - **month** – Integer value representing the month, beginning with 0 for January to 11 for December.
 - **date** – Integer value representing the day of the month.
 - **hour** – Integer value representing the hour of the day 24 – *hourscale*.
 - **minute** – Integer value representing the minute segment of a time reading.
 - **second** – Integer value representing the second segment of a time reading.
 - **millisecond** – Integer value representing the millisecond segment of a time reading.

Date Properties

Here is a list of the properties of the Date object along with their description.

Property	Description
----------	-------------

[constructor](#)

Specifies the function that creates an object's prototype.

[prototype](#)

The prototype property allows you to add properties and methods to an object

In the following sections, we will have a few examples to demonstrate the usage of different Date properties.

Date Methods

Here is a list of the methods used with **Date** and their description.

Method	Description
Date	Returns today's date and time
getDate	Returns the day of the month for the specified date according to local time.
getDay	Returns the day of the week for the specified date according to local time.
getFullYear	Returns the year of the specified date according to local time.
getHours	Returns the hour in the specified date according to local time.
getMilliseconds	Returns the milliseconds in the specified date according to local time.
getMinutes	Returns the minutes in the specified date according to local time.
getMonth	Returns the month in the specified date according to local time.
getSeconds	Returns the seconds in the specified date according to local time.
getTime	Returns the numeric value of the specified date as the number of milliseconds since January 1, 1970, 00:00:00 UTC.
getTimezoneOffset	Returns the time-zone offset in minutes for the current locale.
getUTCDate	Returns the day <i>date</i> of the month in the specified date according to universal time.
getUTCDay	Returns the day of the week in the specified date according to universal time.

<u>getUTCFullYear</u>	Returns the year in the specified date according to universal time.
<u>getUTCHours</u>	Returns the hours in the specified date according to universal time.
<u>getUTCMilliseconds</u>	Returns the milliseconds in the specified date according to universal time.
<u>getUTCMinutes</u>	Returns the minutes in the specified date according to universal time.
<u>getUTCMonth</u>	Returns the month in the specified date according to universal time.
<u>getUTCSeconds</u>	Returns the seconds in the specified date according to universal time.
<u>getYear</u>	Deprecated - Returns the year in the specified date according to local time. Use <u>getFullYear</u> instead.
<u>setDate</u>	Sets the day of the month for a specified date according to local time.
<u>setFullYear</u>	Sets the full year for a specified date according to local time.
<u>setHours</u>	Sets the hours for a specified date according to local time.
<u>setMilliseconds</u>	Sets the milliseconds for a specified date according to local time.
<u>setMinutes</u>	Sets the minutes for a specified date according to local time.
<u>setMonth</u>	Sets the month for a specified date according to local time.
<u>setSeconds</u>	Sets the seconds for a specified date according to local time.
<u>setTime</u>	Sets the Date object to the time represented by a number of milliseconds since January 1, 1970, 00:00:00 UTC.
<u>setUTCDate</u>	Sets the day of the month for a specified date according to universal time.
<u>setUTCFullYear</u>	Sets the full year for a specified date according to universal time.
<u>setUTCHours</u>	Sets the hour for a specified date according to universal time.

[setUTCMilliseconds](#)

Sets the milliseconds for a specified date according to universal time.

[setUTCMinutes](#)

Sets the minutes for a specified date according to universal time.

[setUTCMonth](#)

Sets the month for a specified date according to universal time.

[setUTCSeconds](#)

Sets the seconds for a specified date according to universal time.

[setYear](#)

Deprecated - Sets the year for a specified date according to local time. Use setFullYear instead.

[toDateString](#)

Returns the "date" portion of the Date as a human-readable string.

[toGMTString](#)

Deprecated - Converts a date to a string, using the Internet GMT conventions. Use toUTCString instead.

[toLocaleDateString](#)

Returns the "date" portion of the Date as a string, using the current locale's conventions.

[toLocaleFormat](#)

Converts a date to a string, using a format string.

[toLocaleString](#)

Converts a date to a string, using the current locale's conventions.

[toLocaleTimeString](#)

Returns the "time" portion of the Date as a string, using the current locale's conventions.

[toSource](#)

Returns a string representing the source for an equivalent Date object; you can use this value to create a new object.

[toString](#)

Returns a string representing the specified Date object.

[getTimeString](#)

Returns the "time" portion of the Date as a human-readable string.

[toUTCString](#)

Converts a date to a string, using the universal time convention.

[valueOf](#)

Returns the primitive value of a Date object.

Converts a date to a string, using the universal time convention.

Date Static Methods

In addition to the many instance methods listed previously, the Date object also defines two static methods. These methods are invoked through the Date constructor itself.

Method	Description
Date.parse	Parses a string representation of a date and time and returns the internal millisecond representation of that date.
Date.UTC	Returns the millisecond representation of the specified UTC date and time.

In the following sections, we will have a few examples to demonstrate the usages of Date Static methods

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