Table of Contents

schools
Tutorials ▾ References ▾ Exercises ▾ Spaces Menu ▾

Pro Get Certified Create Website

JavaScript Date Formats

❮ Previous Next ❯

JavaScript Date Input

There are generally 3 types of JavaScript date input formats:

Type	Example
ISO Date	"2015-03-25" (The International Standard)
Short Date	"03/25/2015"
Long Date	"Mar 25 2015" or "25 Mar 2015"

Dark mode

The ISO format follows a strict standard in JavaScript.

The other formats are not so well defined and might be browser specific.

JavaScript Date Output

Independent of input format, JavaScript will (by default) output dates in full text string format:

```
Sat Feb 11 2023 19:33:46 GMT+0600 (Bangladesh Standard Time)
```

JavaScript ISO Dates

ISO 8601 is the international standard for the representation of dates and times.

The ISO 8601 syntax (YYYY-MM-DD) is also the preferred JavaScript date format:

Example (Complete date)

```
const d = new Date("2015-03-25");
```

Dark mode

The computed date will be relative to your time zone.
Depending on your time zone, the result above will vary between March 24 and March 25.

ISO Dates (Year and Month)

ISO dates can be written without specifying the day (YYYY-MM):

Example

```
const d = new Date("2015-03");
```

Try it Yourself »

Dark mode

ISO Dates (Only Year)

ISO dates can be written without month and day (YYYY):

Example

```
const d = new Date("2015");
```

Try it Yourself »

Time zones will vary the result above between December 31 2014 and January 01 2015.

ISO Dates (Date-Time)

ISO dates can be written with added hours, minutes, and seconds (YYYY-MM-DDTHH:MM:SSZ):

Example

Try it Yourself »

Date and time is separated with a capital T.

UTC time is defined with a capital letter Z.

If you want to modify the time relative to UTC, remove the Z and add +HH:MM or -HH:MM instead:

Example

```
const d = new Date("2015-03-25T12:00:00-06:30");
```

Try it Yourself »

UTC (Universal Time Coordinated) is the same as GMT (Greenwich Mean Time).

Omitting T or Z in a date-time string can give different results in different browsers.

When setting a date, without specifying the time zone, JavaScript will use the browser's time zone.

When getting a date, without specifying the time zone, the result is converted to the browser's time zone.

In other words: If a date/time is created in GMT (Greenwich Mean Time), the date/time will be converted to CDT (Central US Daylight Time) if a user browses from central US.

JavaScript Short Dates.

Short dates are written with an "MM/DD/YYYY" syntax like this:

Example

```
const d = new Date("03/25/2015");
```

Try it Yourself »

WARNINGS !

In some browsers, months or days with no leading zeroes may produce an error:

The behavior of "YYYY/MM/DD" is undefined.
Some browsers will try to guess the format. Some will return NaN.

```
const d = new Date("2015/03/25");
```

The behavior of "DD-MM-YYYY" is also undefined.
Some browsers will try to guess the format. Some will return NaN.

```
const d = new Date("25-03-2015");
```

JavaScript Long Dates.

Long dates are most often written with a "MMM DD YYYY" syntax like this:

Example

```
const d = new Date("Mar 25 2015");
```

Month and day can be in any order:

Example

```
const d = new Date("25 Mar 2015");
```

Try it Yourself »

And, month can be written in full (January), or abbreviated (Jan):

Example

```
const d = new Date("January 25 2015");
```

Try it Yourself »

Example

```
const d = new Date("Jan 25 2015");
```

Commas are ignored. Names are case insensitive:

Example

```
const d = new Date("JANUARY, 25, 2015");
```

Try it Yourself »

Date Input - Parsing Dates

If you have a valid date string, you can use the `Date.parse()` method to convert it to milliseconds.

`Date.parse()` returns the number of milliseconds between the date and January 1, 1970:

Example

```
let msec = Date.parse("March 21, 2012");
```

Try it Yourself »

Dark mode

```
arse("March 21, 2012");
e(msec);
```

avaScript Date Reference

reference, go to our:

Date Reference.

ns descriptions and examples of all Date properties and methods.

Next ❯

Dark mode

schools
Tutorials ▾ References ▾ Exercises ▾ Spaces Menu ▾

Pro Get Certified Create Website

HTML CSS JAVASCRIPT SQL

JavaScript Get Date Methods

❮ Previous Next ❯

The new Date() Constructor

In JavaScript, date objects are created with `new Date()`.

`new Date()` returns a date object with the current date and time.

Get the Current Time

```
const date = new Date();
```

Try it Yourself »

Date Get Methods

Method	Description
getFullYear()	Get **year** as a four digit number (yyyy)
getMonth()	Get **month** as a number (0-11)
getDate()	Get **day** as a number (1-31)
getDay()	Get **weekday** as a number (0-6)
getHours()	Get **hour** (0-23)
getMinutes()	Get **minute** (0-59)
getSeconds()	Get **second** (0-59)
getMilliseconds()	Get **millisecond** (0-999)
getTime()	Get **time** (milliseconds since January 1, 1970)

Note 1

The get methods above return **Local time**.

Universal time (UTC) is documented at the bottom of this page.

Note 2

The get methods return information from existing date objects.

In a date object, the time is static. The "clock" is not "running".

The time in a date object is NOT the same as current time.

The getFullYear() Method

The `getFullYear()` method returns the year of a date as a four digit number:

Examples

```
const d = new Date("2021-03-25");
d.getFullYear();
```

Try it Yourself »

```
const d = new Date();
d.getFullYear();
```

Try it Yourself »

Warning !

Old JavaScript code might use the non-standard method getYear().

getYear() is supposed to return a 2-digit year.

getYear() is deprecated. Do not use it!

The getMonth() Method

The `getMonth()` method returns the month of a date as a number (0-11).

Note

In JavaScript, January is month number 0, February is number 1, ...

Finally, December is month number 11.

Examples

```
const d = new Date("2021-03-25");
d.getMonth();
```

Try it Yourself »

```
const d = new Date();
d.getMonth();
```

Try it Yourself »

Note

You can use an array of names to return the month as a name:

Examples

```
const months = ["January", "February", "March", "April", "May", "June", "July", "August", "September",
"October", "November", "December"];

const d = new Date("2021-03-25");
let month = months[d.getMonth()];
```

Try it Yourself »

```
const months = ["January", "February", "March", "April", "May", "June", "July", "August", "September",
"October", "November", "December"];

const d = new Date();
let month = months[d.getMonth()];
```

Try it Yourself »

The getDate() Method

The `getDate()` method returns the day of a date as a number (1-31):

Examples

```
const d = new Date("2021-03-25");
d.getDate();
```

Try it Yourself »

```
const d = new Date();
d.getDate();
```

Try it Yourself »

The getHours() Method

The `getHours()` method returns the hours of a date as a number (0-23):

Examples

```
const d = new Date("2021-03-25");
d.getHours();
```

Try it Yourself »

```
const d = new Date();
d.getHours();
```

Try it Yourself »

The getMinutes() Method

The `getMinutes()` method returns the minutes of a date as a number (0-59):

Examples

```
const d = new Date("2021-03-25");
d.getMinutes();
```

Try it Yourself »

```
const d = new Date();
d.getMinutes();
```

Try it Yourself »

The getSeconds() Method

The `getSeconds()` method returns the seconds of a date as a number (0-59):

Examples

```
const d = new Date("2021-03-25");
d.getSeconds();
```

Try it Yourself »

```
const d = new Date();
d.getSeconds();
```

Try it Yourself »

The getMilliseconds() Method

The `getMilliseconds()` method returns the milliseconds of a date as a number (0-999):

Examples

```
const d = new Date("2021-03-25");
d.getMilliseconds();
```

Try it Yourself »

```
const d = new Date();
d.getMilliseconds();
```

Try it Yourself »

The getDay() Method

The `getDay()` method returns the weekday of a date as a number (0-6).

Note

In JavaScript, the first day of the week (day 0) is Sunday.

Some countries in the world consider the first day of the week to be Monday.

Examples

```
const d = new Date("2021-03-25");
d.getDay();
```

Try it Yourself »

```
const d = new Date();
d.getDay();
```

Try it Yourself »

Note

You can use an array of names, and `getDay()` to return weekday as a name:

Examples

```
const days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];

const d = new Date("2021-03-25");
let day = days[d.getDay()];
```

Try it Yourself »

```
const days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];

const d = new Date();
let day = days[d.getDay()];
```

Try it Yourself »

The getTime() Method

The `getTime()` method returns the number of milliseconds since January 1, 1970:

Examples

```
const d = new Date("1970-01-01");
d.getTime();
```

Try it Yourself »

```
const d = new Date("2021-03-25");
d.getTime();
```

Try it Yourself »

```
const d = new Date();
d.getTime();
```

Try it Yourself »

The Date.now() Method

`Date.now()` returns the number of milliseconds since January 1, 1970.

Examples

```
let ms = Date.now();
```

Try it Yourself »

Calculate the number of years since 1970/01/01:

```
const minute = 1000 * 60;
const hour = minute * 60;
const day = hour * 24;
const year = day * 365;

let years = Math.round(Date.now() / year);
```

Try it Yourself »

`Date.now()` is a static method of the Date object.

You cannot use it on a date object like `myDate.now()` .

The syntax is always `Date.now()` .

w3schools

Tutorials ▾ References ▾ Exercises ▾ Spaces Menu ▾

Pro Get Certified Create Website

JavaScript Set Date Methods

❮ Previous Next ❯

Set Date methods let you set date values (years, months, days, hours, minutes, seconds, milliseconds) for a Date Object.

Set Date Methods

Set Date methods are used for setting a part of a date:

Method	Description

Dark mode

setFullYear()	Set the year (optionally month and day)
setHours()	Set the hour (0-23)
setMilliseconds()	Set the milliseconds (0-999)
setMinutes()	Set the minutes (0-59)
setMonth()	Set the month (0-11)
setSeconds()	Set the seconds (0-59)
setTime()	Set the time (milliseconds since January 1, 1970)

The setFullYear() Method

The `setFullYear()` method sets the year of a date object. In this example to 2020:

Example

```
const d = new Date();
d.setFullYear(2020);
```

Try it Yourself »

The `setFullYear()` method can **optionally** set month and day:

Example

```
const d = new Date();
d.setFullYear(2020, 11, 3);
```

Try it Yourself »

The setMonth() Method

The `setMonth()` method sets the month of a date object (0-11):

Dark mode

```
const d = new Date();
d.setMonth(11);
```

Try it Yourself »

The setDate() Method

The `setDate()` method sets the day of a date object (1-31):

Example

```
const d = new Date();
d.setDate(15);
```

Try it Yourself »

The `setDate()` method can also be used to **add days** to a date:

Example

Dark mode

```
d.setDate(d.getDate() + 30);
```

Try it Yourself »

If adding days shifts the month or year, the changes are handled automatically by the Date object.

The setHours() Method

The `setHours()` method sets the hours of a date object (0-23):

Example

```
const d = new Date();
d.setHours(22);
```

Try it Yourself »

The setMinutes() Method

Dark mode

Example

```
const d = new Date();
d.setMinutes(30);
```

Try it Yourself »

The setSeconds() Method

The `setSeconds()` method sets the seconds of a date object (0-59):

Example

```
const d = new Date();
d.setSeconds(30);
```

Try it Yourself »

Compare Dates

The following example compares today's date with January 14, 2100:

Example

```
let text = "";
const today = new Date();
const someday = new Date();
someday.setFullYear(2100, 0, 14);

if (someday > today) {
  text = "Today is before January 14, 2100.";
} else {
  text = "Today is after January 14, 2100.";
}
```

Try it Yourself »

JavaScript counts months from 0 to 11. January is 0. December is 11.

Complete JavaScript Date Reference

For a complete Date reference, go to our:

Dark mode

ns descriptions and examples of all Date properties and methods.

lf With Exercises

ıte method to set the year of a date object to 2020.

```
v Date();
         ;
```

❮ Previous

Next ❯

Dark mode