JavaScript Date Objects

JavaScript **Date Object** lets us work with dates:

Sat Jan 29 2022 17:05:55 GMT+0100 (hora estándar de Europa central)

Year: 2022 Month: 1 Day: 29 Hours: 17 Minutes: 5

Seconds: 55

Example

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

Try it Yourself »

JavaScript Date Output

By default, JavaScript will use the browser's time zone and display a date as a full text string:

Sat Jan 29 2022 17:05:55 GMT+0100 (hora estándar de Europa central)

You will learn much more about how to display dates, later in this tutorial.

Creating Date Objects

Date objects are created with the `new Date()` constructor.

There are **4 ways** to create a new date object:

```
new Date()
new Date(year, month, day, hours, minutes, seconds, milliseconds)
new Date(milliseconds)
new Date(date string)
```

new Date()

`new Date()` creates a new date object with the **current date and time**:

Example

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

Try it Yourself »

Date objects are static. The computer time is ticking, but date objects are not.

new Date(*year, month, ...*)

`new Date(year, month, ...)` creates a new date object with a **specified date and time**.

7 numbers specify year, month, day, hour, minute, second, and millisecond (in that order):

Example

```
const d = new Date(2018, 11, 24, 10, 33, 30, 0);
```

Try it Yourself »

Note: JavaScript counts months from **0** to **11**:

January = 0.

December = 11.

Specifying a month higher than 11, will not result in an error but add the overflow to the next year:

Specifying:

```
const d = new Date(2018, 15, 24, 10, 33, 30);
```

Is the same as:

```
const d = new Date(2019, 3, 24, 10, 33, 30);
```

Try it Yourself »

Specifying a day higher than max, will not result in an error but add the overflow to the next month:

Specifying:

```
const d = new Date(2018, 5, 35, 10, 33, 30);
```

Is the same as:

```
const d = new Date(2018, 6, 5, 10, 33, 30);
```

Try it Yourself »

Using 6, 4, 3, or 2 Numbers

6 numbers specify year, month, day, hour, minute, second:

Example

```
const d = new Date(2018, 11, 24, 10, 33, 30);
```

Try it Yourself »

5 numbers specify year, month, day, hour, and minute:

Example

```
const d = new Date(2018, 11, 24, 10, 33);
```

Try it Yourself »

4 numbers specify year, month, day, and hour:

Example

```
const d = new Date(2018, 11, 24, 10);
```

Try it Yourself »

3 numbers specify year, month, and day:

Example

```
const d = new Date(2018, 11, 24);
```

Try it Yourself »

2 numbers specify year and month:

Example

```
const d = new Date(2018, 11);
```

Try it Yourself »

You cannot omit month. If you supply only one parameter it will be treated as milliseconds.

Example

```
const d = new Date(2018);
```

Try it Yourself »

Previous Century

One and two digit years will be interpreted as 19xx:

Example

```
const d = new Date(99, 11, 24);
```

Try it Yourself »

Example

```
const d = new Date(9, 11, 24);
```

Try it Yourself »

new Date(*dateString*)

`new Date(dateString)` creates a new date object from a **date string**:

Example

```
const d = new Date("October 13, 2014 11:13:00");
```

Try it Yourself »

Date strings are described in the next chapter.

JavaScript Stores Dates as Milliseconds

JavaScript stores dates as number of milliseconds since January 01, 1970, 00:00:00

UTC (Universal Time Coordinated).

Zero time is January 01, 1970 00:00:00 UTC.

Now the time is: **1643472355663** milliseconds past January 01, 1970

new Date(*milliseconds*)

`new Date(milliseconds)` creates a new date object as **zero time plus milliseconds**:

Example

```
const d = new Date(0);
```

Try it Yourself »

01 January 1970 **plus** 100 000 000 000 milliseconds is approximately 03 March 1973:

Example

```
const d = new Date(100000000000);
```

Try it Yourself »

January 01 1970 **minus** 100 000 000 000 milliseconds is approximately October 31 1966:

Example

```
const d = new Date(-100000000000);
```

Try it Yourself »

Example

```
const d = new Date(86400000);
```

Try it Yourself »

One day (24 hours) is 86 400 000 milliseconds.

Date Methods

When a Date object is created, a number of **methods** allow you to operate on it.

Date methods allow you to get and set the year, month, day, hour, minute, second, and millisecond of date objects, using either local time or UTC (universal, or GMT) time.

Date methods and time zones are covered in the next chapters.

Displaying Dates

JavaScript will (by default) output dates in full text string format:

Example

```
Sat Jan 29 2022 17:05:55 GMT+0100 (hora estándar de Europa central)
```

Try it Yourself »

When you display a date object in HTML, it is automatically converted to a string, with the `toString()` method.

Example

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

Try it Yourself »

The `toUTCString()` method converts a date to a UTC string (a date display standard).

Example

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

Try it Yourself »

The `toDateString()` method converts a date to a more readable format:

Example

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

Try it Yourself »

The `toISOString()` method converts a Date object to a string, using the ISO standard format:

Example

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

Try it Yourself »

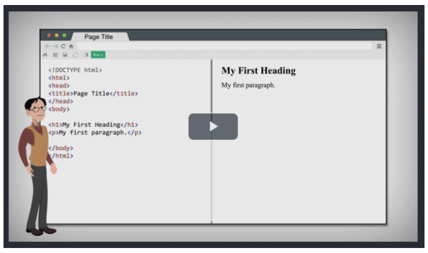
Complete JavaScript Date Reference

For a complete Date reference, go to our:

Complete JavaScript Date Reference.

The reference contains descriptions and examples of all Date properties and methods.

❮ Previous Next ❯



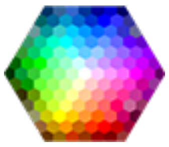
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

My First Heading

My first paragraph.

Explore now

COLOR PICKER



Get certified
by completing
a course today!

Get started

CODE GAME

Play Game

Report Error Forum About Shop

Top Tutorials

HTML Tutorial
CSS Tutorial
JavaScript Tutorial
How To Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
Java Tutorial
C++ Tutorial
jQuery Tutorial

Top References

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
Angular Reference
jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Web Courses

HTML Course
CSS Course
JavaScript Course
Front End Course
SQL Course
Python Course
PHP Course
jQuery Course
Java Course
C++ Course
C# Course
XML Course

Get Certified »