JavaScript toLocaleString()

Back to JS page

Table of Contents

- JavaScript toLocaleString()
  - The toLocaleString() Method
    - Example 1
  - Using Locales
    - Example 2
  - Formatting Currency
    - Example 3
  - Formatting Percentages
    - Example 4
  - Controlling Decimal Digits
    - Example 5
  - Formatting Dates
    - Example 6
  - Date Formatting Options
    - Example 7
  - A Clever Use: Readable File Sizes
    - Example 8
  - Arrays and toLocaleString()
    - Example 9
  - Document
  - Reference

The `toLocaleString()` method converts a value to a string, using local formatting rules. The `toLocaleString()` method is especially useful for formatting numbers, dates, arrays, currencies, and percentages for different countries.

The toLocaleString() Method

The `toLocaleString()` method is available for many JavaScript datatypes / objects:

- Numbers
- Dates
- Arrays
- BigInts

In this example it is used on a Number (`num`):

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript toLocaleString()</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>The toLocaleString() Method</h4>
<p id="demo"></p>

<script>
let num = 1234567.89;

let text = num.toLocaleString();

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 1

Result View Example

Using Locales

You can specify a language and country code to format a value for a specific locale.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript toLocaleString()</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Using Locales</h4>
<p id="demo"></p>

<script>
let num = 1234567.89;

let us = num.toLocaleString("en-US");
let de = num.toLocaleString("de-DE");
let no = num.toLocaleString("no-NO");

document.getElementById("demo").innerHTML =
  "en-US: " + us + "<br>" +
  "de-DE: " + de + "<br>" +
  "no-NO: " + no;
</script>

</body>
</html>
```

Example 2

Result View Example

Formatting Currency

With options, `toLocaleString()` can format numbers as currency.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript toLocaleString()</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Formatting Currency</h4>
<p id="demo"></p>

<script>
let price = 1299.95;

let dollars = price.toLocaleString("en-US",
{style:"currency", currency:"USD"});

let euros = price.toLocaleString("de-DE",
{style:"currency", currency:"EUR"});

let kroner = price.toLocaleString("no-NO",
{style:"currency", currency:"NOK"});

document.getElementById("demo").innerHTML =
  dollars + "<br>" + euros + "<br>" + kroner;
</script>

</body>
</html>
```

Example 3

Result View Example

Formatting Percentages

The `style:"percent"` option formats a number as a percentage.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript toLocaleString()</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Formatting Percentages</h4>
<p id="demo"></p>

<script>
let score = 0.875;

let result = score.toLocaleString("en-US", {style:"percent"});

document.getElementById("demo").innerHTML = result;
</script>

</body>
</html>
```

Example 4

Result View Example

Controlling Decimal Digits

You can control the number of decimal digits with `minimumFractionDigits` and `maximumFractionDigits`.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript toLocaleString()</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Controlling Decimal Digits</h4>
<p id="demo"></p>

<script>
let num = 3.14159;

let text = num.toLocaleString("en-US", {
  minimumFractionDigits: 2,
  maximumFractionDigits: 2
});

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 5

Result View Example

Formatting Dates

Dates can also be formatted with `toLocaleString()`.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript toLocaleString()</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Formatting Dates</h4>
<p id="demo"></p>

<script>
let date = new Date();

let text = date.toLocaleString("en-US");

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 6

Result View Example

Date Formatting Options

Many options can be used to control how dates are displayed.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript toLocaleString()</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Date Formatting Options</h4>
<p id="demo"></p>

<script>
let date = new Date();

let text = date.toLocaleString("en-US", {
  weekday: "long",
  year: "numeric",
  month: "long",
  day: "numeric"
});

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 7

Result View Example

A Clever Use: Readable File Sizes

A clever use of `toLocaleString()` is to make file sizes easier to read.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript toLocaleString()</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>A Clever Use: Readable File Sizes</h4>
<p id="demo"></p>

<script>
function fileSize(bytes) {
  if (bytes < 1024) return bytes + " bytes";
  if (bytes < 1024 * 1024) return (bytes / 1024).toLocaleString("en-US", {maximumFractionDigits:
  return (bytes / 1024 / 1024).toLocaleString("en-US", {maximumFractionDigits: 1}) + " MB";
}

let size = 1536000;
let text = fileSize(size);

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 8

Result View Example

Arrays and toLocaleString()

For arrays, the `toLocaleString()` methods converts each array element.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript toLocaleString()</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Arrays and toLocaleString()</h4>
<p id="demo"></p>

<script>
const dates = [
  new Date("2026-01-01"),
  new Date("2026-12-24")
];

let text = dates.toLocaleString("en-US");

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 9

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript toLocaleString()