

Date in JS



Status

Done

In JavaScript, you can get the current date using the `Date` object. Once you have a `Date` object, you can format it in different ways depending on your requirements. Here's how to get the current date and how to format it in different ways:

1. Getting the Current Date and Time

To get the current date and time:

```
javascriptCopy code
const currentDate = new Date();
console.log(currentDate);
```

This will return something like:

```
2024-09-06T10:14:30.000Z
```

2. Formatting the Date

Here are various ways you can format the current date:

Default ISO Format

```
console.log(currentDate.toISOString());
```

Example output:

```
2024-09-06T10:14:30.000Z
```

The `toISOString()` method converts the date to an ISO string, which is commonly used in databases and APIs.

Locale String Format

You can format the date according to the user's local settings:

```
console.log(currentDate.toLocaleString());
```

Example output:

```
9/6/2024, 10:14:30 AM
```

This uses the system's locale to display the date and time.

Locale Date Only

To display only the date:

```
console.log(currentDate.toLocaleDateString());
```

Example output:

```
9/6/2024
```

Locale Time Only

To display only the time:

```
console.log(currentDate.toLocaleTimeString());
```

Example output:

```
10:14:30 AM
```

Custom Formatting using `toLocaleString`

You can customize how the date is displayed with options:

```
javascriptCopy code
const options = { year: 'numeric', month: 'long', day: 'numeric' };
```

```
console.log(currentDate.toLocaleDateString('en-US', options));
```

Example output:

September 6, 2024

You can customize `options` for different formatting:

- `weekday`: "long", "short", "narrow"
- `year`: "numeric", "2-digit"
- `month`: "numeric", "2-digit", "long", "short", "narrow"
- `day`: "numeric", "2-digit"

Time with Hours and Minutes Only

```
console.log(currentDate.getHours() + ":" + currentDate.getMinutes());
```

Example output:

10:14

Custom Formatting (Manual)

You can manually format the date using various `Date` methods:

```
javascriptCopy code
const year = currentDate.getFullYear();
const month = String(currentDate.getMonth() + 1).padStart(2, '0'); // Months are 0-based
const day = String(currentDate.getDate()).padStart(2, '0');

const formattedDate = `${day}-${month}-${year}`;
console.log(formattedDate);
```

Example output:

```
06-09-2024
```

3. Using Libraries for Advanced Formatting

If you want more control over date formatting, you can use libraries like **Moment.js** or **Day.js**. Here's an example with **Day.js** (a lightweight library similar to Moment.js):

First, install the package:

```
npm install dayjs
```

Then, use it in your code:

```
javascriptCopy code
const dayjs = require('dayjs');
const formatted = dayjs().format('DD-MM-YYYY HH:mm:ss');
console.log(formatted);
```

Example output:

```
06-09-2024 10:14:30
```

Summary of **Date** Methods:

- `getFullYear()` : Returns the year (4 digits).
- `getMonth()` : Returns the month (0-11). Add `+1` for human-readable months.
- `getDate()` : Returns the day of the month (1-31).
- `getDay()` : Returns the day of the week (0-6).
- `getHours()` : Returns the hour (0-23).
- `getMinutes()` : Returns the minutes (0-59).
- `getSeconds()` : Returns the seconds (0-59).