

JavaScript Date Object: Everything You Need to Know

The Date object in JavaScript is used to work with dates and times. It stores dates as the number of milliseconds since January 1, 1970, UTC (also known as Unix Epoch time).

Creating a Date Object

There are four main ways to create a Date object:

1. **Current Date and Time:**

javascript

CopyEdit

```
let myDate = new Date();
```

```
console.log(myDate); // Current date and time
```

- Outputs the current date and time.

2. **Specific Date String:**

javascript

CopyEdit

```
let myCreatedDate = new Date("2025-03-28");
```

```
console.log(myCreatedDate); // Fri Mar 28 2025
```

- The date string should be in the format "YYYY-MM-DD" for reliable results.

3. **Specific Date (Year, Month, Day, Hours, Minutes, Seconds, Milliseconds):**

javascript

CopyEdit

```
let specificDate = new Date(2025, 2, 28, 10, 30, 0); // Month is zero-based (2 = March)
```

```
console.log(specificDate); // Fri Mar 28 2025 10:30:00
```

- Syntax: `new Date(year, month, day, hours, minutes, seconds, milliseconds)`

4. **Milliseconds Since Epoch (Timestamp):**

javascript

CopyEdit

```
let timestampDate = new Date(1679983200000);
```

```
console.log(timestampDate); // Date corresponding to the timestamp
```

- Useful for storing or comparing dates.
-

Getting Timestamps

- Get the current timestamp (milliseconds since January 1, 1970):

javascript

CopyEdit

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

```
console.log(myTimeStamp); // Example: 1679983200000
```

- Get the timestamp from a date object:

javascript

CopyEdit

```
let timeStampFromDate = myCreatedDate.getTime();
```

```
console.log(timeStampFromDate);
```

Date Methods

Getting Date Components

- **Year:**

javascript

CopyEdit

```
console.log(newDate.getFullYear()); // 2025
```

- **Month (0-11):**

javascript

CopyEdit

```
console.log(newDate.getMonth() + 1); // 3 (March)
```

- **Day of the Month (1-31):**

javascript

CopyEdit

```
console.log(newDate.getDate()); // 28
```

- **Day of the Week (0-6, where 0 = Sunday):**

javascript

CopyEdit

```
console.log(newDate.getDay()); // 5 (Friday)
```

- **Hours, Minutes, Seconds, Milliseconds:**

javascript

CopyEdit

```
console.log(newDate.getHours()); // Example: 14  
console.log(newDate.getMinutes()); // Example: 30  
console.log(newDate.getSeconds()); // Example: 45  
console.log(newDate.getMilliseconds()); // Example: 123
```

Formatting Dates

- **String Representation:**

javascript

CopyEdit

```
console.log(myDate.toString()); // Fri Mar 28 2025 14:30:45 GMT+0530 (IST)
```

- **Locale String:**

javascript

CopyEdit

```
console.log(myDate.toLocaleString()); // 28/03/2025, 2:30:45 PM
```

- **Date String Only:**

javascript

CopyEdit

```
console.log(myDate.toDateString()); // Fri Mar 28 2025
```

- **Time String Only:**

javascript

CopyEdit

```
console.log(myDate.toTimeString()); // 14:30:45 GMT+0530 (IST)
```

- **Custom Locale String:**

javascript

CopyEdit

```
console.log(newDate.toLocaleString('en-US', { weekday: 'long' })); // Friday
```

- Options for formatting:

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

- month: "long", "short", "numeric", "2-digit"
 - day: "numeric", "2-digit"
-

Template Literals with Dates

You can combine date methods with template literals for better formatting:

javascript

CopyEdit

```
let dateString = `${newDate.getDay()} and the time  
${newDate.getHours()}:${newDate.getMinutes()}`;  
  
console.log(dateString); // Example: "5 and the time 14:30"
```

Setting Date Components

- **Set Year:**

javascript

CopyEdit

```
newDate.setFullYear(2030);
```

- **Set Month (0-11):**

javascript

CopyEdit

```
newDate.setMonth(5); // June
```

- **Set Day:**

javascript

CopyEdit

```
newDate.setDate(15);
```

- **Set Hours, Minutes, Seconds, Milliseconds:**

javascript

CopyEdit

```
newDate.setHours(10);  
newDate.setMinutes(20);  
newDate.setSeconds(30);  
newDate.setMilliseconds(500);
```



Common Pitfalls and Tips

1. Month Indexing (0-11):

- January is 0, February is 1, ..., December is 11.

2. Locale Differences:

- Date formats can vary based on locale. Use `toLocaleString()` to format according to your locale.

3. Immutable Objects:

- Date objects are **not immutable**. Modifying one can affect other references to it.
-



Example Use Case: Displaying Current Date and Time

javascript

CopyEdit

```
let currentDateTime = new Date();
```

```
console.log(`Today is ${currentDateTime.toDateString()} and the time is  
${currentDateTime.toLocaleTimeString()}`);
```

Output:

pgsql

CopyEdit

Today is Fri Mar 28 2025 and the time is 2:30:45 PM