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" |

JavaScript Date Output

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

Mon Aug 07 2023 17:20:15 GMT+0530 (India 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");

ISO Dates (Year and Month)

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

Example

const d = new Date("2015-03");

ISO Dates (Only Year)

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

Example

const d = new Date("2015");

ISO Dates (Date-Time)

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

Example

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

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");

Time Zones

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");

WARNINGS !

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

const d = new Date("2015-3-25");

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");

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

Example

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

Example

const d = new Date("Jan 25 2015");

Commas are ignored. Names are case insensitive:

Example

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

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");

You can then use the number of milliseconds to **convert it to a date** object:

Example

let msec = Date.parse("March 21, 2012");  
const d = new Date(msec);