Day 6: JavaScript Dates

by AvmnuSng

Dates in JavaScript

Table Of Contents

Date

A JavaScript Date instance represents a single moment in time based on the number of milliseconds elapsed since 1 **January, 1970 UTC**.

Creating Date Instance

There are four constructors we can use to create a *Date* object, defined below.

1. Using `new Date()`

The *default constructor* creates a JavaScript *Date* object for the current date and time (according to your system settings).

2. Using `new Date(value)`

This constructor has a parameter, $value$, which is an integer representing the number of milliseconds elapsed since 1 January 1970 00:00:00 UTC (this is a Unix Epoch, though you should keep in mind that most Unix timestamp functions count in seconds).

3. Using `new Date(dateString)`

This constructor has a parameter, $dateString$, which is a String describing a date. The $dateString$ must be in a format recognized by the `Date.parse()` function, such as `MM/DD/YYYY` or `Month Day, Year`. For example, `01/01/1980` and `Jan 1, 1980` are both strings that can be successfully parsed using the *parse* function.

4. Using new Date(year, month, day, hour, minutes, seconds, milliseconds)

This constructor has the following parameters:

- $year$: An integer denoting the calendar year. Values from 0 through 99 map to the years 1900 through 1999.
- $month$: A one or two digit integer denoting the zero-indexed month. This means that 0 denotes January and 11 denotes December.
- day: Optional. An integer denoting the specific day number within the calendar month.

Go to Top

- $hour$: Optional. An integer denoting the hour of the day.
- $minute$: Optional. An integer denoting the minute segment of a time.
- $second$: Optional. An integer denoting the second segment of a time.
- $millisecond$: Optional. An integer denoting the millisecond segment of a time.

- EXAMPLE

Input Format

The first line contains $date1$.
The second line contains $date2$.

```javascript
'use strict';
process.stdin.on('data', function (data) {
    main(String(data).split("\n"));
});

function main(input) {
    var date1 = new Date(input[0]);
    console.log(date1.toString());
    console.log(date1.toISOString());

    var date2 = new Date(input[1]);
    console.log(date2.toString());
    console.log(date2.toISOString());
```

Input

```
10/11/2009
2016, 04, 22
```

Run

Output

The code above produces the following output:

```
Sun Oct 11 2009 00:00:00 GMT+0000 (UTC)
2009-10-11T00:00:00.000Z
Fri Apr 22 2016 00:00:00 GMT+0000 (UTC)
2016-04-22T00:00:00.000Z
```

Date `get` Methods

1. Date.getTime()

Get the time in milliseconds elapsed since $\text{January } 1, 1970$.

2. Date.getFullYear()

Get the four-digit year $(yyyy)$.

3. Date.getMonth()

Get the *Date* object's month as a zero-indexed number $(0 - 11)$.

4. Date.getDate()

Get the *Date* object's day as a number $(1 - 31)$.

Go to Top

5. Date.getDay()

Get the *Date* object's weekday as a number ($0 - 6$).

6. Date.getHours()

Get the *Date* object's hour ($0 - 23$).

7. Date.getMinutes()

Get the *Date* object's minutes ($0 - 59$)

8. Date.getSeconds()

Get the *Date* object's seconds ($0 - 59$).

9. Date.getMilliseconds()

Get the *Date* object's milliseconds ($0 - 999$).

- EXAMPLE

Click *Run* below to see this in code.

Input Format

A single date string in a format recognized by `Date.parse()`.

```
'use strict';
process.stdin.on('data', function (data) {
    main(String(data));
});
/**** Ignore above this line. ****/

function main(input) {
    let date = new Date(input);

    console.log("date: " + date);
    console.log("date.getDate(): " + date.getDate());
    console.log("date.getDay(): " + date.getDay());
    console.log("date.getFullYear(): " + date.getFullYear());
    console.log("date.getHours(): " + date.getHours());
    console.log("date.getMilliseconds(): " + date.getMilliseconds())
    console.log("date.getMinutes(): " + date.getMinutes());
    console.log("date.getMonth(): " + date.getMonth());
    console.log("date.getSeconds(): " + date.getSeconds());
    console.log("date.getTime(): " + date.getTime());
    console.log("date.getYear(): " + date.getYear());
    console.log("date.toDateString(): " + date.toDateString());
```

Input

```
Feb 3, 1987 12:34:56:789
```

Run

Output

The code above produces the following output:

```
date: Tue Feb 03 1987 12:34:56 GMT+0000 (UTC)
date.getDate(): 3
date.getDay(): 2
date.getFullYear(): 1987
```

Go to Top

```
date.getHours(): 12
date.getMilliseconds(): 789
date.getMinutes(): 34
date.getMonth(): 1
date.getSeconds(): 56
date.getTime(): 539354096789
date.getYear(): 87
date.toDateString(): Tue Feb 03 1987
```

You could also create a date object for the date given as input using the following date constructor and arguments:

```
// Date(year, month, day, hour, minutes, seconds, milliseconds)
let date = new Date(1987, 1, 3, 12, 34, 56, 789);
```

Go to Top