



FREE eBook

LEARNING momentjs

Free unaffiliated eBook created from
Stack Overflow contributors.

#momentjs

Table of Contents

About.....	1
Chapter 1: Getting started with momentjs.....	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Browser.....	2
With local JS file.....	2
With CDN.....	2
Node.js.....	2
Bower.....	3
General usage.....	3
Format Dates.....	3
Relative Time.....	3
Calendar Time.....	3
Multiple Locale Support.....	3
Turn a Date object into and from a moment object.....	4
Calculating and manipulating dates.....	4
Date parsing.....	4
Chapter 2: countdown timer in minutes using moment.js.....	6
Examples.....	6
countdown timer using moment.js.....	6
Credits.....	9

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [momentjs](#)

It is an unofficial and free momentjs ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official momentjs.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with momentjs

Remarks

This section provides an overview of what momentjs is, and why a developer might want to use it.

It should also mention any large subjects within momentjs, and link out to the related topics. Since the Documentation for momentjs is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Moment is a javascript library that was designed to make working with dates and times less time-consuming and more intuitive.

You can easily start using this library by installing it in you web-app by using one of the following guides.

Browser

You can either download the JS file [from the official website](#) or use [cdnjs.com](#).

With local JS file

```
<script src="moment.js"></script>
<script>
  moment().format();
</script>
```

With CDN

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.14.1/moment.min.js"></script>
<script>
  moment().format();
</script>
```

Node.js

```
npm install moment
```

Moment should be included inside the `node_modules` directory, Then add it to your script

```
var moment = require('moment');
moment().format();
```

Bower

```
bower install --save moment
```

To test if your site or web application is properly using the moment.js script, open console (F12 on the keyboard) and type `moment`, the console should print the moment function if it was included.

General usage

The way `Moment` uses dates and times is by wrapping the existing `Date()` object in a `moment()` object and specifying useful and intuitive methods on this object.

Format Dates

```
moment().format('MMMM Do YYYY, h:mm:ss a'); // August 4th 2016, 10:41:45 am
moment().format('dddd'); // Thursday
moment().format('MMM Do YY'); // Aug 4th 16
moment().format('YYYY [escaped] YYYY'); // 2016 escaped 2016
moment().format(); // 2016-08-04T10:41:45+02:00
```

Relative Time

```
moment("201111031", "YYYYMMDD").fromNow(); // 5 years ago
moment("20120620", "YYYYMMDD").fromNow(); // 4 years ago
moment().startOf('day').fromNow(); // 11 hours ago
moment().endOf('day').fromNow(); // in 13 hours
moment().startOf('hour').fromNow(); // 42 minutes ago
```

Calendar Time

```
moment().subtract(10, 'days').calendar(); // 07/25/2016
moment().subtract(6, 'days').calendar(); // Last Friday at 10:42 AM
moment().subtract(3, 'days').calendar(); // Last Monday at 10:42 AM
moment().subtract(1, 'days').calendar(); // Yesterday at 10:42 AM
moment().calendar(); // Today at 10:42 AM
moment().endOf('day'); // Today at 23:59 AM
moment().add(1, 'days').calendar(); // Tomorrow at 10:42 AM
moment().add(3, 'days').calendar(); // Sunday at 10:42 AM
moment().add(10, 'days').calendar(); // 08/14/2016
```

Multiple Locale Support

```
moment.locale();           // en
moment().format('LT');     // 10:43 AM
moment().format('LTS');   // 10:43:14 AM
moment().format('L');     // 08/04/2016
moment().format('l');     // 8/4/2016
moment().format('LL');    // August 4, 2016
moment().format('ll');    // Aug 4, 2016
moment().format('LLL');   // August 4, 2016 10:43 AM
moment().format('lll');   // Aug 4, 2016 10:43 AM
moment().format('LLLL');  // Thursday, August 4, 2016 10:43 AM
moment().format('llll');  // Thu, Aug 4, 2016 10:43 AM
```

Turn a Date object into and from a moment object

To turn a Javascript Date object into a moment (moment object) just call `moment` and pass the Date as a argument

```
moment(new Date());
// Moment {_isAMomentObject: true, _i: Sun Jul 31 2016 11:08:02 GMT+0300 (Jerusalem Daylight
Time), _isUTC: false, _pf: Object, _locale: Locale...}
```

To turn the moment back to a Javascript Date use the `.toDate()` method

```
var now = new Date();
var mom = moment(now)
//undefined
mom
// Moment {_isAMomentObject: true, _i: Sun Jul 31 2016 11:10:32 GMT+0300 (Jerusalem Daylight
Time), _isUTC: false, _pf: Object, _locale: Locale...}
mom.toDate()
// Sun Jul 31 2016 11:10:32 GMT+0300 (Jerusalem Daylight Time)
```

Calculating and manipulating dates

If you have a moment object you can use `add` and `subtract` to manipulate it or set any property of the time directly

```
moment("2016-01-01").add(1, 'year').format('YYYY-MM-DD')
// -> "2017-01-01"
```

Or use `.day()`, `.month()`, `.year()`, `.seconds()`, `.milliseconds()` to set those values directly. Note that the arguments passed are 0-indexed.

```
moment("2016-01-01").month(4).day(5).format('YYYY-MM-DD')
// -> "2016-05-06"
```

Date parsing

You can use moment to parse date strings.

By default moment tries to parse the date as an [ISO-8601](#) string and if that does not work falls back to the browsers `new Date()`. Since the way that browsers construct dates varies it is best to try not to fall back to this.

```
moment('2016-02-04').format('YYYY-MM-DD')  
// ->'2016-02-04'
```

If you have a format other than ISO 8601 it is best to pass the format string as a second argument to moment.

```
moment('04-02-2016', 'MM-DD-YYYY').format('YYYY-MM-DD')  
// ->'2016-02-04'
```

Read [Getting started with momentjs online](https://riptutorial.com/momentjs/topic/1982/getting-started-with-momentjs): <https://riptutorial.com/momentjs/topic/1982/getting-started-with-momentjs>

Chapter 2: countdown timer in minutes using moment.js

Examples

countdown timer using moment.js

HTML

```
<div class="countdown"></div>
```

JS

```
var duration = moment.duration({
  'minutes': 5,
  'seconds': 00

});

var timestamp = new Date(0, 0, 0, 2, 10, 30);
var interval = 1;
var timer = setInterval(function() {
  timestamp = new Date(timestamp.getTime() + interval * 1000);

  duration = moment.duration(duration.asSeconds() - interval, 'seconds');
  var min = duration.minutes();
  var sec = duration.seconds();

  sec -= 1;
  if (min < 0) return clearInterval(timer);
  if (min < 10 && min.length !== 2) min = '0' + min;
  if (sec < 0 && min !== 0) {
    min -= 1;
    sec = 59;
  } else if (sec < 10 && sec.length !== 2) sec = '0' + sec;

  $('.' + 'countdown').text(min + ':' + sec);
  if (min == 0 && sec == 0)
    clearInterval(timer);

}, 1000);
```

Putting it all together

```
<!DOCTYPE html>
<html>
<head>
  <title>FINAL COUNTDOWN</title>
  <style type="text/css">
    /* Styling it all now! */
```



```

@font-face {
  font-family: 'digi';
  src: url('http://cssdeck.com/uploads/resources/fonts/digii/DS-DIGII.TTF');
}

* {
  box-sizing: border-box;
}

html, body {
  width: 100%; height: 100%;
}

body {
  font-family: 'digi';
  font-size: 76%;
  background: #000;
  background-image: radial-gradient(
    center 0,
    white,
    #B9B7B1
  );
}

.countdown {
  width: 250px; height: 100px;
  background: white;
  font-size: 4em;
  color: #707070;

  /* Centering everything */
  position: absolute;
  left: 50%; top: 50%;
  margin-left: -125px; margin-top: -50px;

  text-align: center;
  line-height: 100px;

  border-top: 5px solid #E54B6B;
  border-bottom: 5px solid #E54B6B;
  background-image: linear-gradient(#f0f0f0, white);
}
</style>
<script
src="https://code.jquery.com/jquery-3.2.1.js"
integrity="sha256-DZAnKJ/6XZ9si04Hgrsxu/8s717jcIzLy3oi35EouyE="
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.5.0/moment.min.js"></script>
</head>
<body>
<div class="countdown"></div>
</body>
</html>
<script type="text/javascript">
  var duration = moment.duration({
    'minutes': 5,
    'seconds': 00
  });

```

```
var timestamp = new Date(0, 0, 0, 2, 10, 30);
var interval = 1;
var timer = setInterval(function() {
    timestamp = new Date(timestamp.getTime() + interval * 1000);

    duration = moment.duration(duration.asSeconds() - interval, 'seconds');
    var min = duration.minutes();
    var sec = duration.seconds();

    sec -= 1;
    if (min < 0) return clearInterval(timer);
    if (min < 10 && min.length != 2) min = '0' + min;
    if (sec < 0 && min != 0) {
        min -= 1;
        sec = 59;
    } else if (sec < 10 && sec.length != 2) sec = '0' + sec;

    $('<div>.countdown').text(min + ':' + sec);
    if (min == 0 && sec == 0)
        clearInterval(timer);

}, 1000);

</script>
```

Fiddle : <https://jsfiddle.net/pahday/fyxu66hk/6/>

Read countdown timer in minutes using moment.js online:

<https://riptutorial.com/momentjs/topic/10107/countdown-timer-in-minutes-using-moment-js>

Credits

S. No	Chapters	Contributors
1	Getting started with momentjs	Community , James , k-nut , Molske , Sebastian Witeczek , svarog , VincenzoC
2	countdown timer in minutes using moment.js	Amit Arya