Hey Linda,

I know you’ve been trying to get into Web Development, and to learn JavaScript. Based on my past experience, the best way to learn coding is to do projects.

Coding front-end projects in JavaScript nowadays, it’s handy to use 3rd party libraries. This way you aren’t reinventing the wheel, and using battle tested libraries, that do one thing and one thing well.

One of my favorite libraries is [Moment.js](http://momentjs.com/). It’s a library that helps you work with dates and times. Dates and times, dealing with timezones and such can be tricky in any language, so this library makes things easier.

Moment.js title on it’s homepage says “Parse, validate, manipulate, and display dates and times in JavaScript.” That’s quite a mouthful, let me walk you through each step.

# Parsing Date

“Parse” basically means translating or converting a thing into something else. In this case with Moment.js, parsing dates, you can convert a text string date, into a “moment object”, that you can further validate, manipulate or display in a custom format.

For example turning “10/20/2017” into a Moment object that you display as “October, 20th 2017”

## Using Parse

Moment provides a lot of ways to parse (or create) a moment object.

### Simplelist way to get a moment instance.

The simplest way to get a moment object is simply run `moment()` with no arguments.

```

const m = moment();

```

What this does is simply create a moment instance of the current datetime. Aka the current *moment* at the time of invocation. Now you know how moment got it’s name.

### Another simple way to get Moment instance (not recommended)

Another simple way to get a moment instance is to pass it a string.

```

const m = moment("04-27-2017");

```

Note if you don’t pass a text string that doesn’t match common date formats, either [ISO 8601](https://en.wikipedia.org/wiki/ISO_8601) or [RFC 2822 Date time](https://tools.ietf.org/html/rfc2822#section-3.3), Moment will fallback to `Date.parse()`, and you will receive a warning.

Before version 2.6 released in April 2014, `moment(String)` use to be less strict (and not show the warning) and falls back to `Date.parse` silently. JavaScript Date parse is very quirky. Mozilla foundation says “It is not recommended to use Date.parse as until ES5, parsing of strings was entirely implementation dependent.”, Which is why moment warns us now.

For an [example of it quirks](http://blog.dygraphs.com/2012/03/javascript-and-dates-what-mess.html).

```  
> new Date("2017-04-27").toDateString()

// "Wed Apr 26 2017"

> new Date("2017/04/27").toDateString()

// "Thu Apr 27 2017"

```

Sidebar  
Weird right? Why is it doing this? I have to admit the first time I saw this behavior I was dumbfounded. I thought dates & times were tough, but this is ridiculous.

The reason why `new Date("2017-04-27").toDateString()` produces `Wed Apr 26` and `new Date("2017/04/27").toDateString()` (notice the slashes instead of hyphens) produces `Thur Apr 27` is one of the quirks of JavaScript.

JavaScript is over 20 years old and initially created in only 10 days by Brendan Eich of Netscape. Which is an amazing feat. Since it’s wildly popular, there are some [quirks](http://developer.telerik.com/featured/seven-javascript-quirks-i-wish-id-known-about/).

Some of these quirks are getting address with newer editions of JavaScript (like `=> arrow functions` for this context quirk) but since JavaScript is so popular it’ll be unwise to introduce breaking changes.

So because “2017-04-27” is in ISO format, Date is parsing it as UTC, whereas “2017/04/27” is not ISO format, so it get’s parse as local time.

So when I create and print back `new Date(“2017-04-27”).toDateString()`, I’m creating a Date instance of UTC datetime of 2017-04-27 00:00:00, which is equivalent to my local datetime 2017-04-26 20:00:00”, then when I print this Date instance in local time, it says it’s Wed Apr 26, which it is in London, but I wanted to create a Date instance for the datetime here, not Dublin (Not taking daylights savings into account).

Read more about this in the “**Differences in assumed time zone” in** [**Mozilla docs**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/parse)**.**

### Recommended way to parse text strings

The recommended way to create moment instances with date strings is to [pass a format](https://momentjs.com/docs/#/parsing/string-format/).

```

// moment(String, Format)

Const m = moment(“04/27/2017”, “MM/DD/YYYY”);

```

List of format tokens can be found [here](https://momentjs.com/docs/#/displaying/format/).

# Validate Dates

# Manipulate Dates

# Display Dates

----

Maybe article should be about, Selling Moment.js

1. Datetime are hard, strange and full of quirks.
2. Simple / elegant API.
3. Powerful Manipulation