

Riverside JS Workshop *isDate()* Code Challenge

7/3/2016

The Challenge

Build a cross-app, cross-browser JavaScript method that will take a text string argument and reliably determine whether the string represents a parse-able date. It should provide fewer false positives than native JS `Date.parse()` and be less restrictive than `moment.js` using a date format in strict mode.

You will find the [get the code here](#).

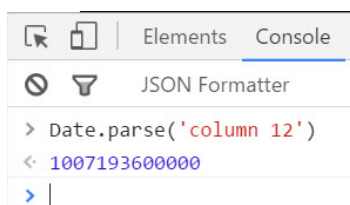
Some Background

Working with JavaScript dates is tricky. In particular, trusting user input or data from an API can lead to bad results. For many, it's standard practice to check a date string before relying on it. One common practice has been to use the `Date.parse()` to verify that you have an actual date.

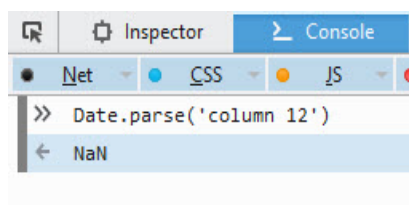
```
if(Date.parse(date_input)){
    //date_input is a parseable date so do something with it
    ...
}
```

Unfortunately, this technique no longer returns reliable results across all browsers. Google Chrome no longer returns reliable results for many values that aren't dates. Take the string, `column 12`, for example. Let's see how three major browsers handle this input using `Date.parse()`—*running on Windows 10 Professional*.

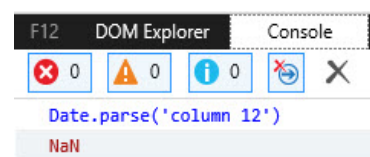
Chrome 51.0.2704.103



Firefox 45.0.1



Microsoft Edge 25.10586.0.0



Limitations of Moment.js

To get reliable date checking across browsers, a third party library seems a logical choice – enter [Moment.js](#). Moment's solution is pretty good, but it requires some careful configuration. By default, it merely wraps the native JS's `Date.parse()` and returns similar results. As of version 2.13.0, this method has been deprecated, and your browser will tell you so.

Moment does allow for the application of date format filters, such as [ISO 8601](#), as well a strict mode that requires precise format including dashes and slashes. This is reliable but strict – very strict, requiring a careful per-app configuration.

Finding a Better Solution

This challenge asks you to build a reliable solution that will work across browsers and across applications. It should not require user configuration. Likewise it should both eliminate false positives that occur in native JS and support a wider range of dates than moment.js.

Share your Solution

When you have a working solution to share, post a link in the Issues section of the [RiversideJSW Repository](#). Good luck, and happy coding!