

How to survive this new JavaScript fragmentation era

by Andrea Giammarchi
@WebReflection

A brief history about me

A brief history about me

- full stack Web development since 00's

A brief history about me

- full stack Web development since 00's
- Certified Zend (PHP) Engineer, Macromedia AS1/AS2 Developer

A brief history about me

- full stack Web development since 00's
- Certified Zend (PHP) Engineer, Macromedia AS1/AS2 Developer
- developing under ECMAScript 3rd edition for both JS and AS1

A brief history about me

- full stack Web development since 00's
- Certified Zend (PHP) Engineer, Macromedia AS1/AS2 Developer
- developing under ECMAScript 3rd edition for both JS and AS1
- developing under ECMAScript 4th Edition for AS2 time frame

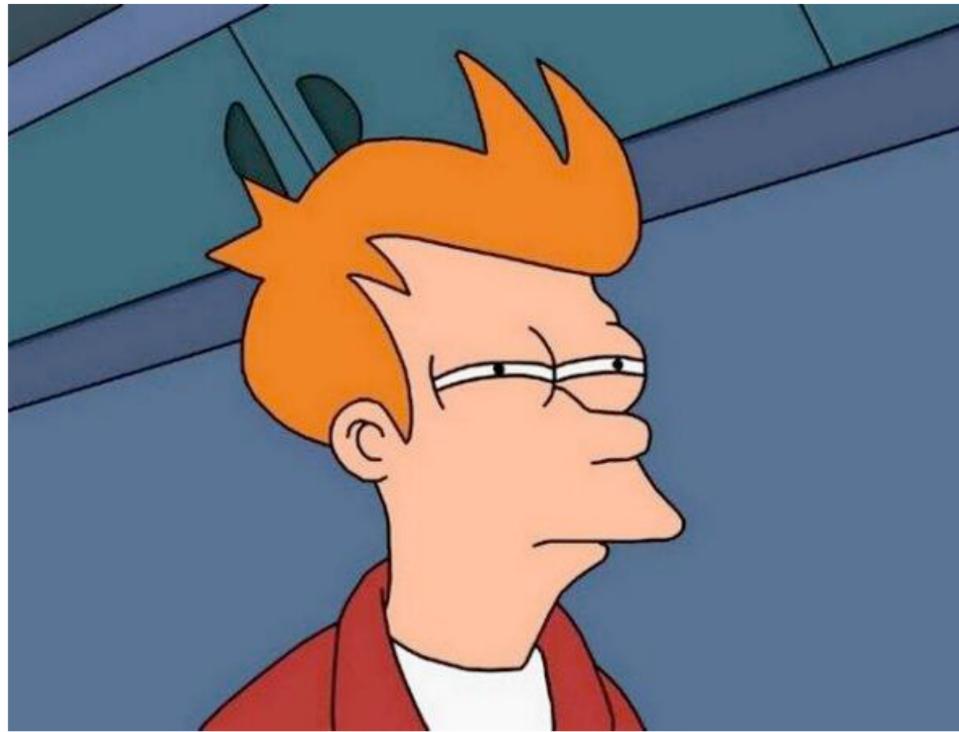
2003 ActionScript 2

```
class Greeter {  
    var greeting:String;  
    function Greeter(message:String) {  
        this.greeting = message;  
    }  
    function greet(Void) :String {  
        return "Hello, " + this.greeting;  
    }  
}  
var greeter:Greeter = new Greeter("world");
```

... 9 years later ...

2012 TypeScript

```
class Greeter {  
    greeting:String;  
    constructor(message:String) {  
        this.greeting = message;  
    }  
    greet(Void):String {  
        return "Hello, " + this.greeting;  
    }  
}  
var greeter:Greeter = new Greeter("world");
```



A brief history about me

- full stack Web development since 00's
- Certified Zend (PHP) Engineer, Macromedia AS1/AS2 Developer
- developing under ECMAScript 3rd edition for both JS and AS1
- developing under ECMAScript 4th Edition for AS2 time frame

A brief history about me

- full stack Web development since 00's
- Certified Zend (PHP) Engineer, Macromedia AS1/AS2 Developer
- developing under ECMAScript 3rd edition for both JS and AS1
- developing under ECMAScript 4th Edition for AS2 time frame
- ditching Flash for Open Web Standards

A brief history about me

- full stack Web development since 00's
- Certified Zend (PHP) Engineer, Macromedia AS1/AS2 Developer
- developing under ECMAScript 3rd edition for both JS and AS1
- developing under ECMAScript 4th Edition for AS2 time frame
- ditching Flash for Open Web Standards
- developing under ECMAScript 5th Edition (sort of)

ECMAScript 5th Edition



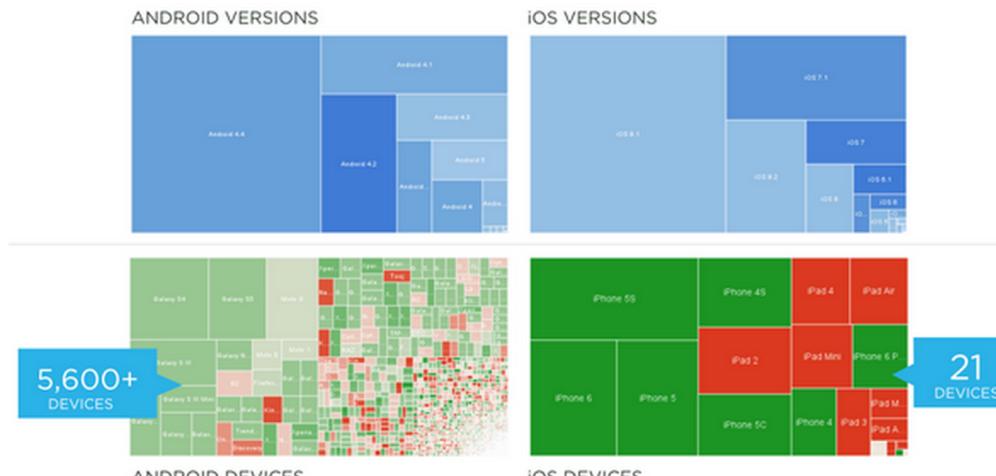
Luke Wroblewski

@lukew



Following

the same but different.
scientiamobile.com/page/movr-mobi...



ECMAScript 5th Edition



Jeffrey Zeldman 

@zeldman



Following

“Samsung shits out a device every 30 seconds.” @bradfrost

#RWD #mobile #design #aeabos



...

RETWEETS

19

FAVORITES

35



6:23 AM - 13 May 2015

A brief history about me

- full stack Web development since 00's
- Certified Zend (PHP) Engineer, Macromedia AS1/AS2 Developer
- developing under ECMAScript 3rd edition for both JS and AS1
- developing under ECMAScript 4th Edition for AS2 time frame
- ditching Flash for Open Web Standards
- developing under ECMAScript 5th Edition (sort of)

A brief history about me

- full stack Web development since 00's
- Certified Zend (PHP) Engineer, Macromedia AS1/AS2 Developer
- developing under ECMAScript 3rd edition for both JS and AS1
- developing under ECMAScript 4th Edition for AS2 time frame
- ditching Flash for Open Web Standards
- developing under ECMAScript 5th Edition (sort of)
- developing under ECMAScript 5th + 5.1 + 6 (aka ES2015) + 7

ES2015



A brief history about me

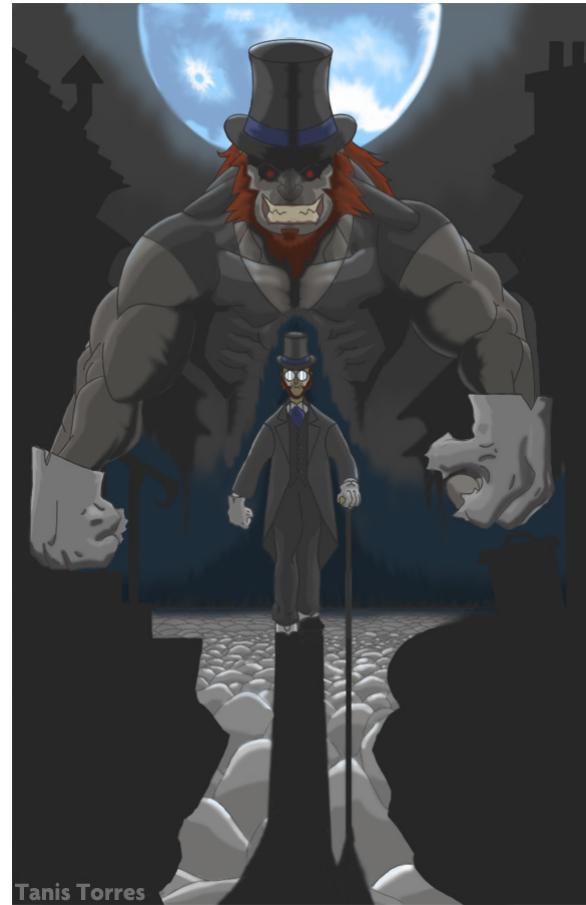
- full stack Web development since 00's
- Certified Zend (PHP) Engineer, Macromedia AS1/AS2 Developer
- developing under ECMAScript 3rd edition for both JS and AS1
- developing under ECMAScript 4th Edition for AS2 time frame
- ditching Flash for Open Web Standards
- developing under ECMAScript 5th Edition (sort of)
- developing under ECMAScript 5th + 5.1 + 6 (aka ES2015) + 7
- developing under ES 3, 5, 5.1, 6 / 2015, 7 / 20XX

Transpilers

Transpilers

a new hope ?

Transpilers



Transpilers

- could change code logic and behavior without warnings

Transpilers

- could change code logic and behavior without warnings
- could target different engines than ours

Transpilers

- could change code logic and behavior without warnings
- could target different engines than ours
- could have partial support compromising a standard adoption

Transpilers

- could change code logic and behavior without warnings
- could target different engines than ours
- could have partial support compromising a standard adoption
- could compromise standards due early adoption

Transpilers

- could change code logic and behavior without warnings
- could target different engines than ours
- could have partial support compromising a standard adoption
- could compromise standards due early adoption



Domenic Denicola
@domenic



[Follow](#)

If you are using Babel, please provide feedback on the new function bind syntax proposal! It needs your support!
babeljs.io/blog/2015/05/1...

📍 Manhattan, NY



Transpilers

2015-03-15

Using the ES6 transpiler Babel on Node.js

Labels: [dev](#), [esnext](#), [javascript](#)

This blog post explains how to use [the ES6 transpiler Babel](#) with Node.js. You can [download the code shown in this post](#) on GitHub. For further information on ECMAScript 6, consult the ebook “[Exploring ES6](#)”.

Warning: The approach explained in this post is convenient for experiments and development. But it uses on-the-fly transpilation, which may be too slow for your production code. Then you can transpile as a build step ([as explained in the Babel documentation](#)).

<http://www.2ality.com/2015/03/babel-on-node.html>

Transpilers

It's time to make the switch.

Babel is available wherever you use JavaScript - so start using it today!

Transpilers

It's time to make the switch.

Babel is available wherever you use JavaScript - so start using it today!

.... or ... IS IT ???



Allen Wirfs-Brock (2 hours ago)

Reply

[View Original](#)

[Go to Source](#)

[Edit](#)

On Jun 2, 2015, at 8:08 PM, Logan Smyth wrote:

To clarify things, since I don't think it's been made abundantly clear, the example that Sebastian gave would work in a standard ES6 environment, correct? It is only if the callback were executed synchronously that the exception would be thrown since the `this` binding has not yet been initialized? Transpilers however have elected to prevent this to err on the side of ensuring that invalid ES6 allowed through because adding runtime checking for the `this` binding would be difficult?

In other words, transpilers have elected to be buggy. ECMAScript 2015 does not give implementations an option in this regard. If an implementation produces such an early error it is not in compliance with the standard.

Transpilers

- very good as experimental playground

Transpilers

- very good as experimental playground
- could be used in production ****only**** if you have tested all your target browsers

Transpilers

- very good as experimental playground
- could be used in production ****only**** if you have tested all your target browsers
- should not derail from standards

Transpilers

- very good as experimental playground
- could be used in production ****only**** if you have tested all your target browsers
- should not derail from standards
- should not adopt too early

Transpilers

- very good as experimental playground
- could be used in production ****only**** if you have tested all your target browsers
- should not derail from standards
- should not adopt too early
- Should probably warn every time a feature is not fully implemented or incompatible with some engine

Transpilers

Most of them suffer JS fragmentation:

Know your real targets,
and test all the things!

What about Polyfills ?

What about Polyfills ?

- instead of syntax, polyfills “transpile” features

What about Polyfills ?

- instead of syntax, polyfills “transpile” features
- watch out incomplete features (`Object.create`)

What about Polyfills ?

- instead of syntax, polyfills “transpile” features
- watch out incomplete features (`Object.create`)
- watch out your real targets: always test, rarely trust!

What about Polyfills ?



WebReflection commented on 59f400f 14 days ago



Using this link: <http://goo.gl/kNOpQm>

Android 2.3 has 7 failures

Function apply, doesn't work with array like objects. Arguments list has wrong type

String split /.?/ results in []

String split /.??/ results in []

String split /()/ results in []

Object isExtensible doesn't return false if object is not extensible

Object isExtensible doesn't return false if object is seal

Object isExtensible doesn't return false if object is freeze (frozen?)

Android 4.0.4 has 2 failures

Function apply, doesn't work with array like objects. Arguments list has wrong type

bind does not return the value that instance of original class when called as a constructor

iOS 5.1 has 2 failures

Array forEach apparently does autobox the content in strict mode

Function apply, doesn't work with array like objects. [object Object] is not a valid argument

webOS 2 has 2 failures (it's WebKit)

Function apply, doesn't work with array like objects. Arguments list has wrong type

bind does not return the value that instance of original class when called as a constructor

What about Polyfills ?

BB 7 has 11 failures (WebKit)

Function apply, doesn't work with array like objects. [object Object] is not a valid argument

String '..split /()() results in [.]

String '..split /(.??)(.??)/ results in [.]

String tesst.split(/(s)/) results in [t, undefined, 'e', 's', 't']

String tesst.split(/(s)?/) results in [t, undefined, 'e', undefined, 's', undefined, 't']

String tesst.split(/(s)/) results in [t, 'e', 'ss', 't']

String tesst.split(/(s?)/) results in [t, " , 'e' , " , 's' , " , 's' , " , 't']

String '..split(/((((.??))))/) results in [.]

Object isExtensible doesn't return false if object is not extensible

Object isExtendible doesn't return false if object is seal

Object isExtendible doesn't return false if object is freeze (frozen?)

Opera Mini for Android 2.3 or 4.4 has 3 failures

String 'test'.split(/(?:)/, 'two') results in []

Number should have NaN ... Expected -2147483648 not to be -2147483648

Date toJSON return null for not finite dates

I couldn't find the version of Opera that was causing problems, so I guess that comment can go now.

UC Browser regular and UC Browser Mini work fine.

What about Polyfills ?

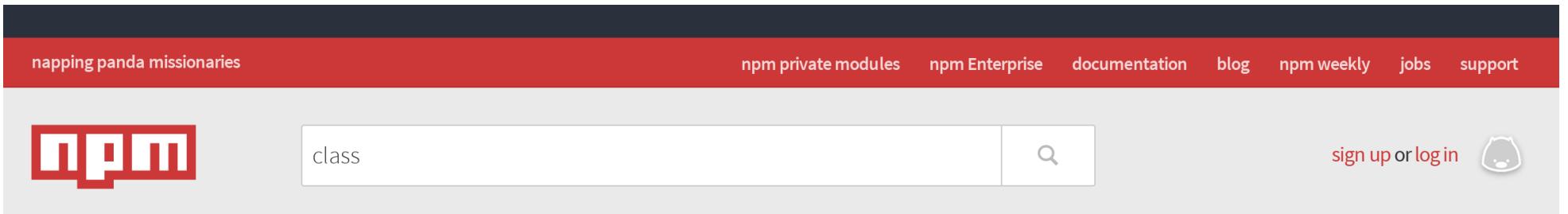
```
<!DOCTYPE html>
<html>
  <head>
    <title>Modern DOM</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=0,minimal-ui">
    <!--[if lte IE 9]><script>(function(f){window.setTimeout=f(window.setTimeout);window.setInterval=f(window.setInterval)})(function(f){return function(c,t){var a=[].slice.call(arguments,2);return f(function(){c.apply(this,a)},t)}});</script>
    <![endif]-->
    <!--[if IE 8]><script src="//cdnjs.cloudflare.com/ajax/libs/ie8/0.2.3/ie8.js"></script><![endif]-->
    <script src="//cdnjs.cloudflare.com/ajax/libs/dom4/1.3.1/dom4.js"></script>
    <!--[if IE 8]><script src="//cdnjs.cloudflare.com/ajax/libs/document-register-element/0.3.0/dre-ie8-upfront-fix.js">
    </script><![endif]-->
    <script src="//cdnjs.cloudflare.com/ajax/libs/document-register-element/0.3.0/document-register-element.js"></script>
    <!--[if IE 8]>
      <script src="//cdnjs.cloudflare.com/ajax/libs/es5-shim/4.1.0/es5-shim.js"></script>
      <script src="//cdnjs.cloudflare.com/ajax/libs/es5-shim/4.1.0/es5-sham.js"></script>
    <![endif]-->
  </head>
  <body>
  </body>
</html>
```

Not a NIH Syndrome

Not a NIH Syndrome

NIH => Not Invented Here

Not a NIH Syndrome



The screenshot shows the npm search interface. At the top, there's a red header bar with the text "napping panda missionaries". Below it is a navigation bar with links: "npm private modules", "npm Enterprise", "documentation", "blog", "npm weekly", "jobs", and "support". The main area features the "npm" logo on the left and a search bar containing the word "class". To the right of the search bar is a magnifying glass icon. Further right are links for "sign up or log in" and a small user icon. The search results title "9453 results for 'class'" is centered above a grid of project cards.



class

A simple yet powerful Ruby-like Class inheritance sys...

version 0.1.4

16 downloads in the last week



is-class

Check if function is an ES6 class.

version 0.0.3

4 downloads in the last week



extend-class

Creates a new class inheriting the methods and prop...

version 1.0.4

0 downloads in the last week



class-registry

class registry

version 0.0.1

0 downloads in the last week



define-js-class

define js class

version 0.0.2

5 downloads in the last week



rework-class-prefix

Add a class prefix to further sandbox CSS styling.

version 1.0.0

7 downloads in the last week

Not a NIH Syndrome



Reginald Braithwaite
@raganwald



Following

ES7 decorators solve a problem that ES6 classes create.

github.com/wycats/javascript-decorators



...



GitHub

[wycats/javascript-decorators](https://github.com/wycats/javascript-decorators)

Contribute to javascript-decorators development by creating an account on GitHub.



[View on web](#)

Not a NIH Syndrome



Tom
@phuunet



Following

This. Is. Insanity. Please. Stop.

github.com/wycats/javascript-decorators



...



GitHub

wycats/javascript-decorators

Contribute to javascript-decorators development by creating an account on GitHub.



[View on web](#)

Not a NIH Syndrome

es-class

build passing

A future proof, backward compatible, JavaScript class utility.

- **ES6** and **ES7** friendly, through semantic `extends` and `constructor` plus other features
- **ES5** natively compatible granting that what you write is what you get
- **ES3** backward compatible, thanks to common minifiers that will make production code safe

Feel free to [test all features](#) related to this project, and please come back if your browser is not green.

Not a NIH Syndrome

Desktop

- IE 6 or greater
- Firefox 3 and greater
- Chrome
- Safari
- Opera

Not a NIH Syndrome

Desktop

- IE 6 or greater
- Firefox 3 and greater
- Chrome
- Safari
- Opera

Mobile

- Android 2 or greater
- iOS 5 or greater
- IE9 Mobile or greater
- Bada
- Kindle Fire
- Xpress
- Opera Mini and Mobile
- Blackberry
- Ubuntu Phone

Not a NIH Syndrome

Desktop

- IE 6 or greater
- Firefox 3 and greater
- Chrome
- Safari
- Opera

Mobile

- Android 2 or greater
- iOS 5 or greater
- IE9 Mobile or greater
- Bada
- Kindle Fire
- Xpress
- Opera Mini and Mobile
- Blackberry
- Ubuntu Phone

Others (tested)

- [Espruino Pico](#)
- MIPS version of node for [Arduino Yun](#)
- node.js 0.6+ and io.js every (so Raspberry PI and every other with node.js)
- Duktape and Nashorn

... back to fragmentation ...

Browsers, W3C and WHATWG

Browsers, W3C and WHATWG

- vendors are running to implement any sort of new feature

Browsers, W3C and WHATWG

- vendors are running to implement any sort of new feature
- few engines lack core features behind (ES6 Proxy)

Browsers, W3C and WHATWG

- vendors are running to implement any sort of new feature
- few engines lack core features behind (ES6 Proxy)
- few features get out incomplete (Fetch API)

Browsers, W3C and WHATWG

- vendors are running to implement any sort of new feature
- few engines lack core features behind (ES6 Proxy)
- few features get out incomplete (Fetch API)
- the quicker some specific vendor pushes an API or feature out, the more fragmentation we have

Browsers, W3C and WHATWG

- vendors are running to implement any sort of new feature
- few engines lack core features behind (ES6 Proxy)
- few features get out incomplete (Fetch API)
- the quicker some specific vendor pushes an API or feature out, the more fragmentation we have
- every browser update makes it actually less “One JS” than ever (welcome to the evergreen-maybe era)

Browsers, W3C and WHATWG

- vendors are running to implement any sort of new feature
- few engines lack core features behind (ES6 Proxy)
- few features get out incomplete (Fetch API)
- the quicker some specific vendor pushes an API or feature out, the more fragmentation we have
- every browser update makes it actually less “One JS” than ever (welcome to the evergreen-maybe era)
- the Web today is optimized for Chrome, like it was in late 90's

Browsers, W3C and WHATWG

- learn standards and use widely tested polyfills
OR
- hope your framework of choice got it right in terms of gracefully enhanced compatibility and overall performance

Browsers, W3C and WHATWG

- learn standards and use widely tested polyfills
OR
- hope your framework of choice got it right in terms of gracefully enhanced compatibility and overall performance

Wednesday, April 22, 2015

Is Anyone Testing On Mobile?

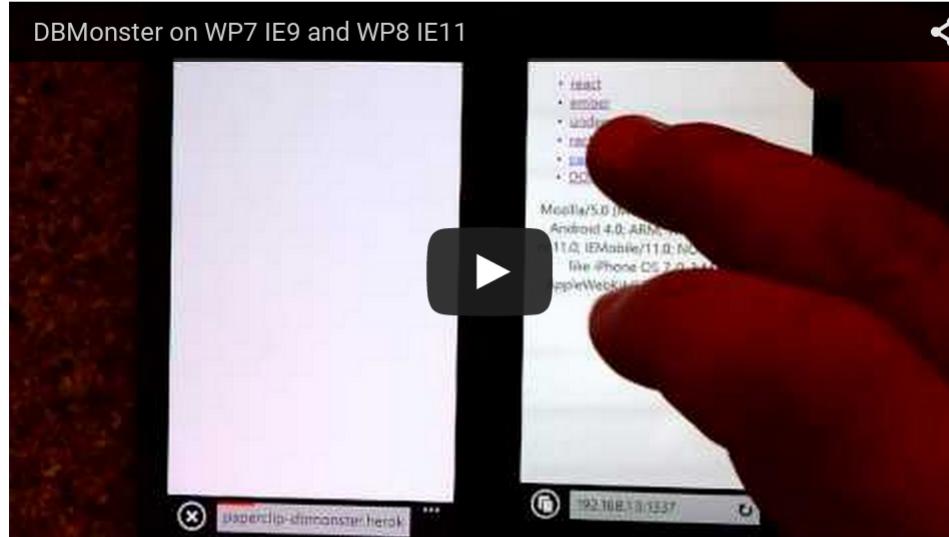
<http://webreflection.blogspot.co.uk/2015/04/is-anyone-testing-on-mobile.html>

Browsers, W3C and WHATWG

Mobile IE9 and IE11

React here never appears in IE9, and it performs like 1 frame per each 3 seconds performance. Worst than pre-alpha Ember, and less compatible.

To be fair, every other bench performed crap except for paperclip and DOM. It's hard to tell who won here, 'cause why I have the feeling paperclip is slightly faster on IE11, it doesn't do so well in IE9 Mobile.



Do Not Assume It Will Work As Expected

Do Not Assume It Will Work As Expected



MAY 17, 2015

How to test React
components using Karma and
webpack

Do Not Assume It Will Work As Expected

Dependencies

You'll need to install various packages. It looks like a lot of dependencies, but all the non-Karma packages will be necessary for general module bundling during development.

Full set of required packages:

Do Not Assume It Will Work As Expected

Dependencies

You'll need to install various packages. It looks like a lot of dependencies, but all the non-Karma packages will be necessary for general module bundling during development.

Full set of required packages:

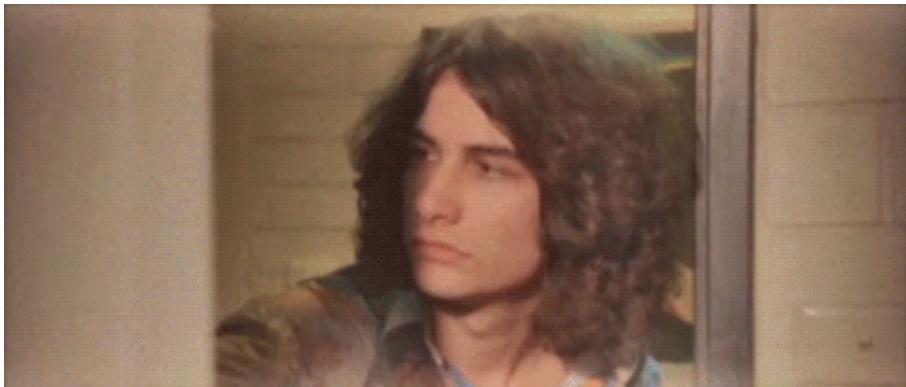
- [babel-core](#): to transform JSX and transpile ES6.
- [babel-loader](#): webpack loader for Babel.
- [karma](#): the test runner.
- [karma-chai-plugins](#): assertion library (or your library of choice).
- [karma-chrome-launcher](#): Chrome browser launcher.
- [karma-cli](#): CLI tool for Karma.
- [karma-mocha](#): adapter for the Mocha testing framework.
- [karma-sourcemap-loader](#): sourcemap loader.
- [karma-webpack](#): to use webpack with Karma.
- [react](#)
- [webpack](#): module bundler

Do Not Assume It Will Work As Expected

Dependencies

You'll need to install various packages. It looks like a lot of dependencies, but all the non-Karma packages will be necessary for general module bundling during development.

Full set of required packages:



- [babel-core](#): to transform JSX and transpile ES6.
- [babel-loader](#): webpack loader for Babel.
- [karma](#): the test runner.
- [karma-chai-plugins](#): assertion library (or your library of choice).
- [karma-chrome-launcher](#): Chrome browser launcher.
- [karma-cli](#): CLI tool for Karma.
- [karma-mocha](#): adapter for the Mocha testing framework.
- [karma-sourcemap-loader](#): sourcemap loader.
- [karma-webpack](#): to use webpack with Karma.
- [react](#)
- [webpack](#): module bundler

Server Side JS

Server Side JS

- io.js

Server Side JS

- io.js
- NodeJS 0.12.X

Server Side JS

- io.js
- NodeJS 0.12.X
- NodeJS 0.10.X (MIPS, Edison)

Server Side JS

- io.js
- NodeJS 0.12.X
- NodeJS 0.10.X (MIPS, Edison)
- Nashorn (Java 8 Rhino replacement)

Server Side JS

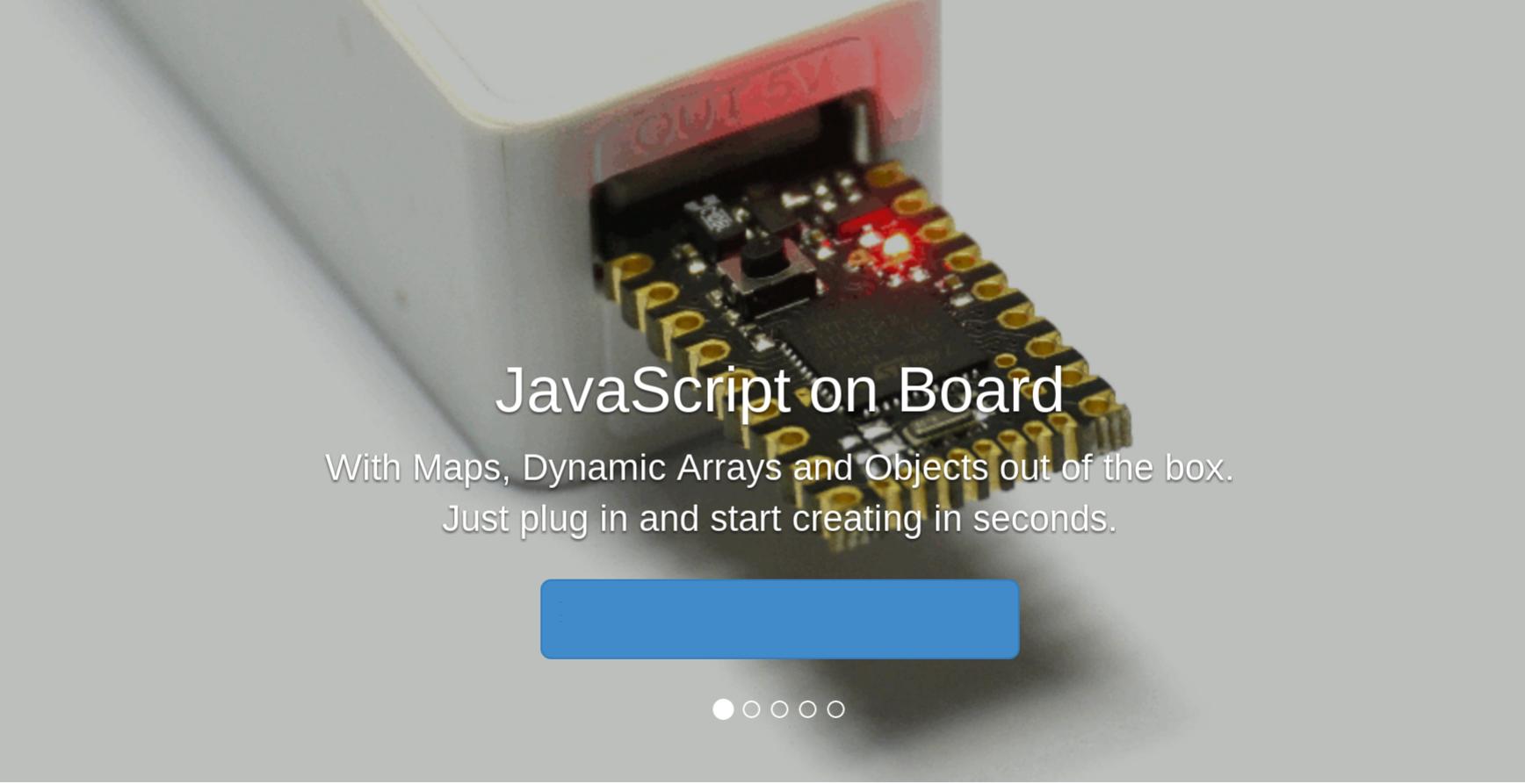
- io.js
- NodeJS 0.12.X
- NodeJS 0.10.X (MIPS, Edison)
- Nashorn (Java 8 Rhino replacement)
- Rhino (RingoJS + others)

Server Side JS

- io.js
- NodeJS 0.12.X
- NodeJS 0.10.X (MIPS, Edison)
- Nashorn (Java 8 Rhino replacement)
- Rhino (RingoJS + others)
- Duktape (with LibUV)

JS and the Internet of Things

JS and the Internet of Things



JavaScript on Board

With Maps, Dynamic Arrays and Objects out of the box.
Just plug in and start creating in seconds.



● o o o o

JS and the Internet of Things

Screenshot of a GitHub repository page for "Espruino JavaScript".

The page includes the following elements:

- Header:** GitHub logo, search bar ("Search GitHub"), navigation links ("Explore", "Gist", "Blog", "Help"), user profile ("WebReflection") with a picture, and various icons for account management.
- Repository Overview:** A large image of a white breadboard, the title "Espruino JavaScript" with a "fork" icon, a brief description ("Espruino is a JavaScript interpreter for microcontrollers. It can fit into devices with as little as 128kB Flash and 8kB RAM."), and contact information ("http://www.espruino.com" and "gw@espruino.com").
- Filters and Search:** "Filters" dropdown and "Find a repository..." search bar.
- People:** A sidebar section titled "People" showing one user entry: "gfwilliams" (Gordon Williams) with a profile picture of a breadboard.
- Repository Details:** The main content area shows the repository "Espruino" with the description "The Espruino JavaScript interpreter - Official Repo", updated "6 hours ago", and metrics: 432 stars, 130 forks, and a commit history chart.

JS and the Internet of Things

A screenshot of the npm search interface. The top navigation bar features the npm logo, a search input field containing "morse", a search icon, and links for "sign up or log in".

morse

sign up or log in

37 results for '**morse**'



morse

Morse code for mad scientists

version 0.1.0

7 downloads in the last week



morse-node

Encode strings to Morse code, decode strings from M...

version 0.1.1

10 downloads in the last week



morjs

Library for encoding/decoding Morse code messages

version 1.1.0

4 downloads in the last week



morse-stream

transform words into morse code

version 0.0.0

3 downloads in the last week



tessel-morse

Communicate in Morse code using your Tessel

version 0.1.0

2 downloads in the last week



ancient-morse-translator

Just a fun brain teaser

version 0.0.0

7 downloads in the last week



morse-code

A morse code input library

version 0.1.0

3 downloads in the last week



morse-beep

A simple text to Morse Code player for the modern web

version 1.1.0

10 downloads in the last week



morsecode

Turn your alphanumeric content into Morse code.

version 1.0.0

2 downloads in the last week

JS and the Internet of Things

Q: How many engineers does it take to change a light bulb?

A1: None. They are all too busy trying to design the perfect light bulb.

A2: Only the one with the instruction manual.

A3: One. But she would insist that the way she did it was distinctive.

A4: Three. One to hold the ladder, one to hold the light bulb, and the third to interpret the Japanese text.

A5: Five. One to design a nuclear-powered light bulb that never needs changing, one to figure out how to power the rest of the USA using that nuked light bulb, two to install it, and one to write the computer program that controls the wall switch.

A6: None. "According to my calculations, the problem doesn't exist."

As Summary

As Summary

 **Horse JS**
@horse_js

Following

inclined to be severely prejudiced against anything that mentions "isomorphic"

← ↗ ⭐ ...

RETWEET	FAVORITES
1	4

6:11 AM - 19 May 2015

 Reply to @horse_js

 **isomorphicJS** @isomorphicJS · 16s
@horse_js agreed

← ↗ ⭐ 1 +1 ...



As Summary

As Summary

- ES3 syntax is still the safest bet

As Summary

- ES3 syntax is still the safest bet
- ES5 descriptors just became widely usable across platforms

As Summary

- ES3 syntax is still the safest bet
- ES5 descriptors just became widely usable across platforms
- ES6 is OK on Web but watch out real-world Production targets

As Summary

- ES3 syntax is still the safest bet
- ES5 descriptors just became widely usable across platforms
- ES6 is OK on Web but watch out real-world Production targets
- forget for now ES6/7 on IoT (slower, too much RAM, poor support)

As Summary

- ES3 syntax is still the safest bet
- ES5 descriptors just became widely usable across platforms
- ES6 is OK on Web but watch out real-world Production targets
- forget for now ES6/7 on IoT (slower, too much RAM, poor support)
- **learn vanilla JavaScript**, you'll rarely regret it, and you'll be always ready to go

How to survive this new JavaScript fragmentation era

Thank You!

@WebReflection