



LEARNING JAVASCRIPT WITHOUT NODE.JS

Jakub Marchwicki @kubem

DISCLAIMER

This is a bit JavaScript 101

You've been warned



1995 - THE YEAR JS BORN



1995 - THE YEAR JS BORN

In 10 days





NOKIA

Plus





mrWizard



Computer



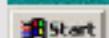
My Network
Places



Recycle Bin



Netscape
Naviga...



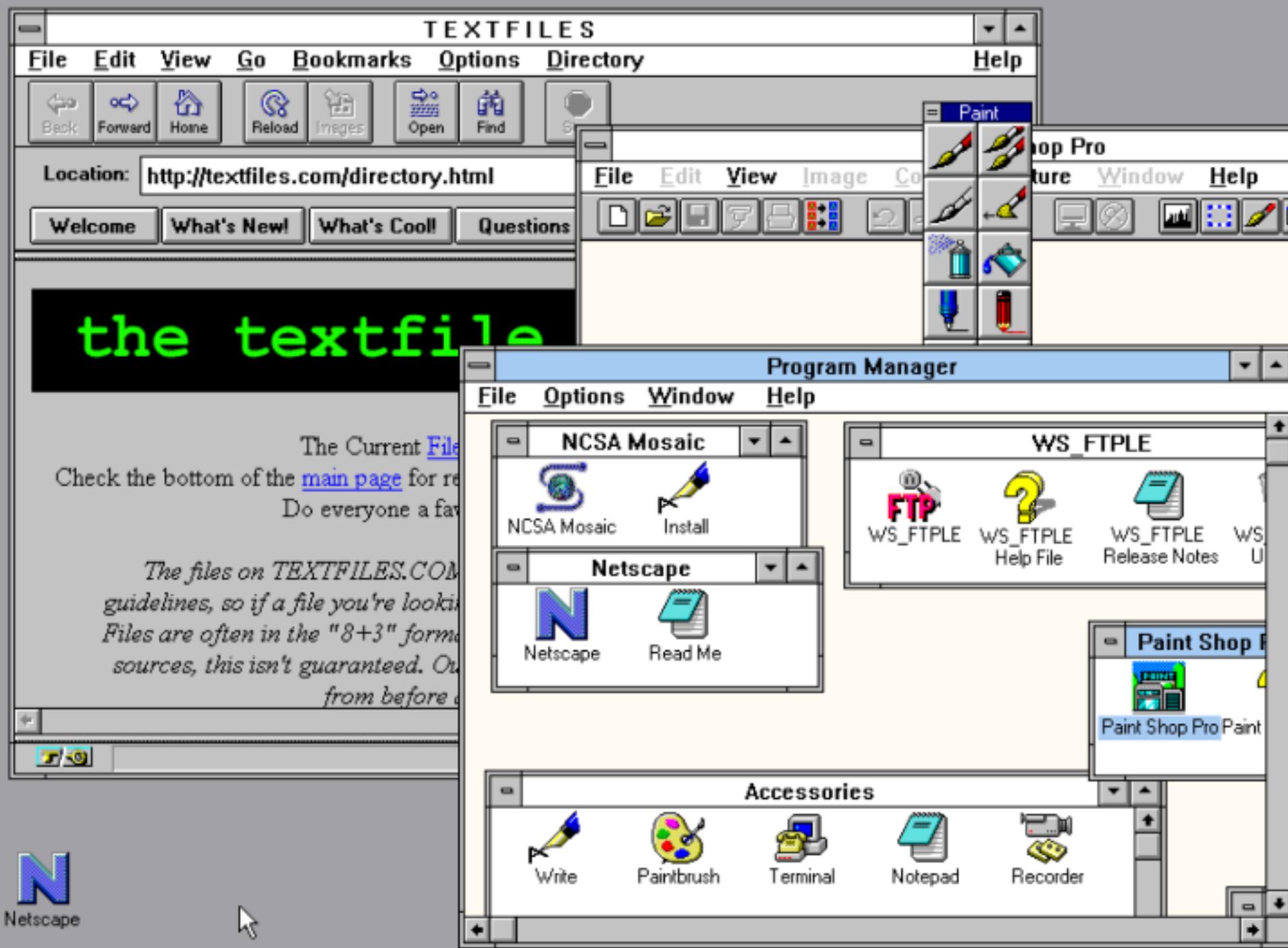
Sketchy Icons by ~Azur...

13. Red Hot Chilli Pepper...

C:\Documents and Settin...

Jasc Paint Shop Pro - Wi...

1:21 PM





mrWizard



Computer



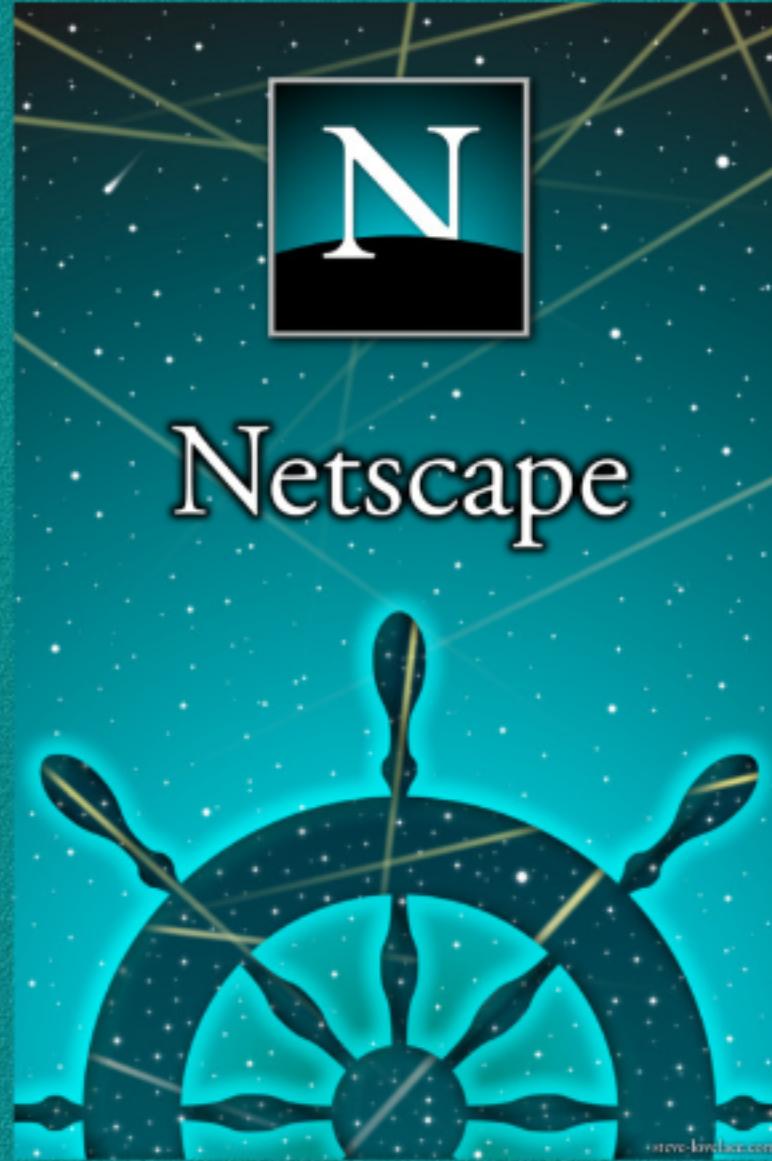
My Network
Places



Recycle Bin



Netscape
Naviga...



Start

Sketchy Icons by ~Azur...

13. Red Hot Chilli Pepper...

C:\Documents and Settin...

Jasc Paint Shop Pro - W...

1:21 PM



What's
On
Today

12/24/96
15:30 EST

Today's
News on the
Net

Today's
PhotoWire

Corporate
News on the
Net

Electronic
Media Kit

Today's High
Tech News
on the Net

Today's
Health News
on the Net

Corporate
Profile on the
Net

Personal Web
Box

Tradeshow
News

Retail Reports

Cool Web
Sites

ABOUT BW/
< >



Dow Jones **Enter**
Business Directory

Today's News
on the Net
click for full screen



Enter words describing a concept or keywords you wish to find information about:

SEARCH

This Search only covers news releases from the past 48 hours

[Click here for news from the previous 24 hours](#)

15:00 Headlines

- [Interpool to pay cash dividends on common stock](#)
- [Interwest Home Medical Inc. announces annual revenues and income and two acquisitions in Colorado](#)
- [Bridgford Foods Corp. announces record sales and \\$5.7 million in earnings for the 1996 fiscal year](#)
- [Golden Books completes sale of Penn Corporation to Peacock Papers](#)
- [Silver Standard Acquires Large Silver Resource in Australia](#)
- [20th Century Insurance Co. to renew homeowner policies](#)
- [Consolidated Westview - Private Placement Financing](#)

14:00 Headlines

- [Computer Marketplace announces commencement of private placement](#)
- [Prostate Cancer Mortality Rate Decreases for African-American Males, but Mortality Increases for Other Races Overall](#)
- [Diversified Investors sells TNPI subsidiary to American Bank Note for up to \\$675,000 cash, stock, payout](#)
- [DENTSPLY International Inc. and New Image Industries sign Letter of Intent](#)
- [Ican Minerals - Racey Porphyry Gold Deposit, Oregon](#)

13:00 Headlines

- [2544-3482 Quebec Inc. - Announcement](#)
- [Reserve Royalty And Stateside Oil Announce Acquisition](#)
- [■ ADVISORY/Photo from TV series "Brotherly Love" available on BW PhotoWire/AP PhotoExpress, Presslink and Business Wire's Home Page on the World Wide Web](#)
- [159585 Canada Inc. - Announcement](#)
- [Board of Directors authorized reduction in conversion price of company's Class A Common Stock Purchase Warrants](#)

THE STATE OF AN ART JAVASCRIPT

As of 1995

```
var today = new Date();
var dd = today.getDate();
var mm = today.getMonth()+1; //January is 0!
var yyyy = today.getFullYear();

if(dd<10) {
    dd='0'+dd
}

if(mm<10) {
    mm='0'+mm
}

today = mm+'/' +dd+ '/' +yyyy;
document.write('<blink>' + today + '</blink>');
```



IN ONE SENTENCE

*JavaScript on the client side, sprinkled with lots
of “if this is Netscape do this, else do that”
because back then noone had any idea how to
use JavaScript properly.*

<https://medium.freecodecamp.com/being-a-developer-after-40-3c5dd112210c>

— Adrian Kosmaczewski



NOT WHAT WE WANT TO LEARN



Nat Pryce

@natpryce

Static typing is really helpful ... unless the only type used is String



RETWEETS

34

FAVORITES

22

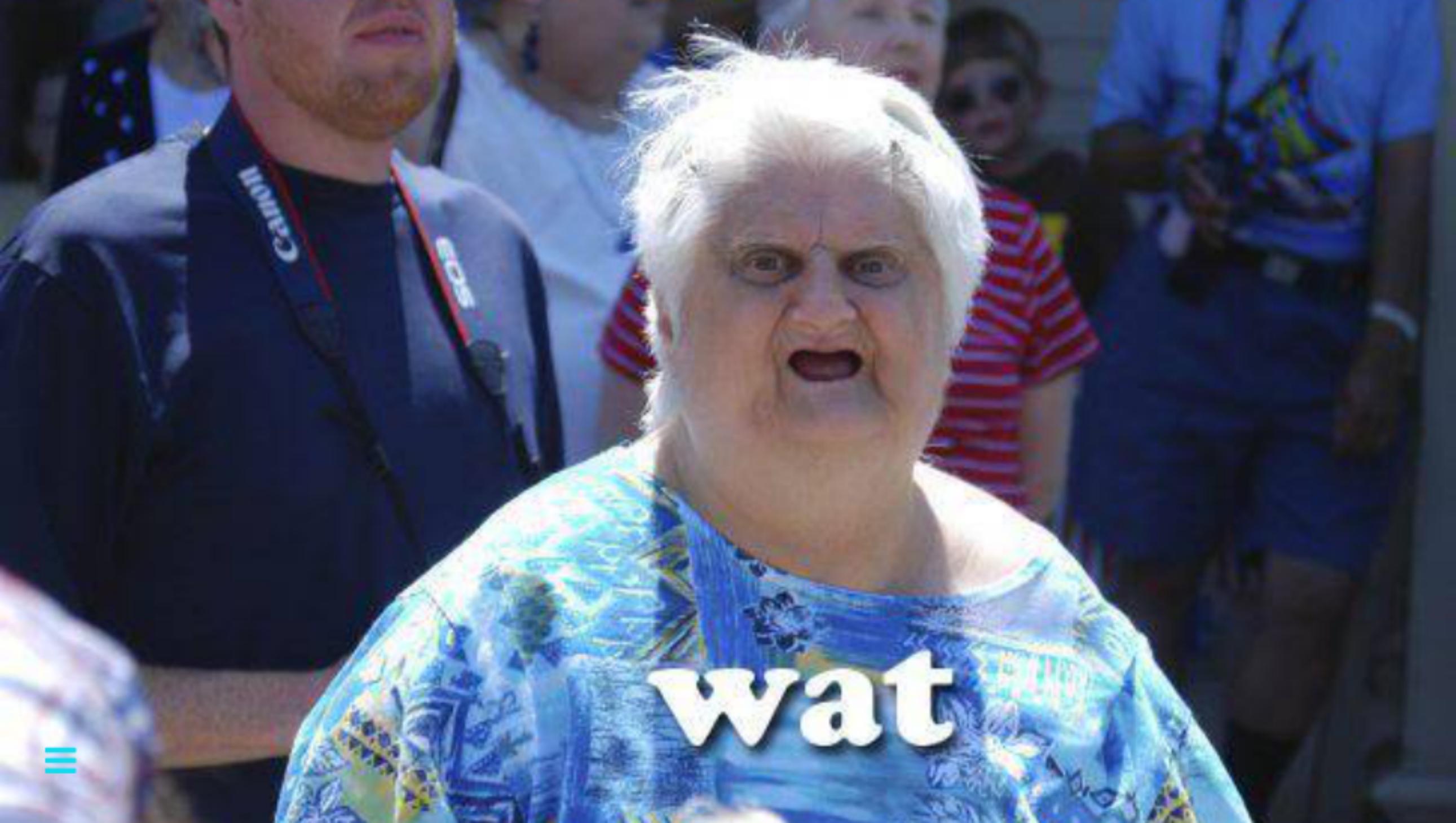


7:02 AM - 20 Jan 2015

<http://twitter.com/natpryce/status/557553705109950464>



Jakub Marchwicki @kubem



wat



Unearthing the Excellence



JavaScript: The Good Parts



YAHOO! PRESS

Douglas Crockford





Any application that can be written in JavaScript,
will eventually be written in JavaScript.

Jeff Atwood, 2007

<http://blog.codinghorror.com/the-principle-of-least-power/>



CONTENT

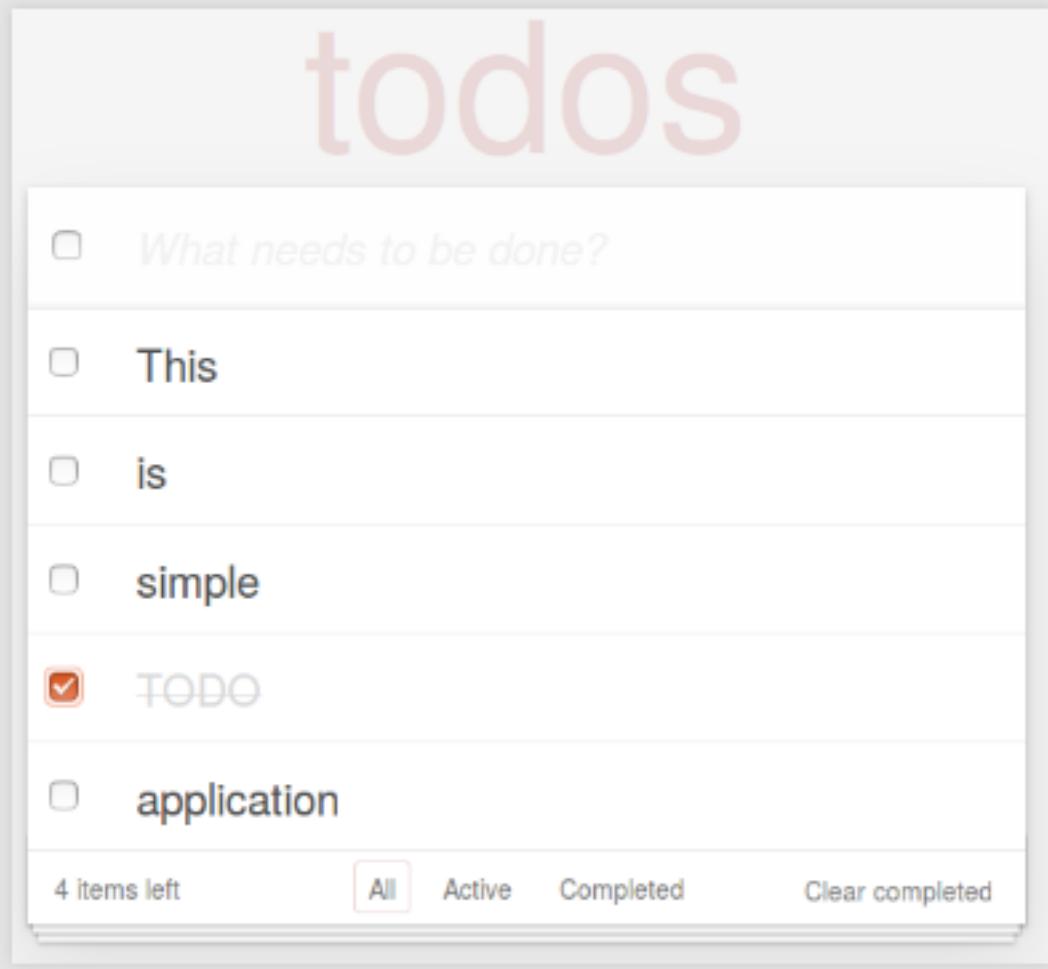
- JavaScript Basics & Gotchas
- Modern framework bullsh*t bingo game
- Fighting language inconsistencies



WHO AM I?



TODOMVC EXAMPLES



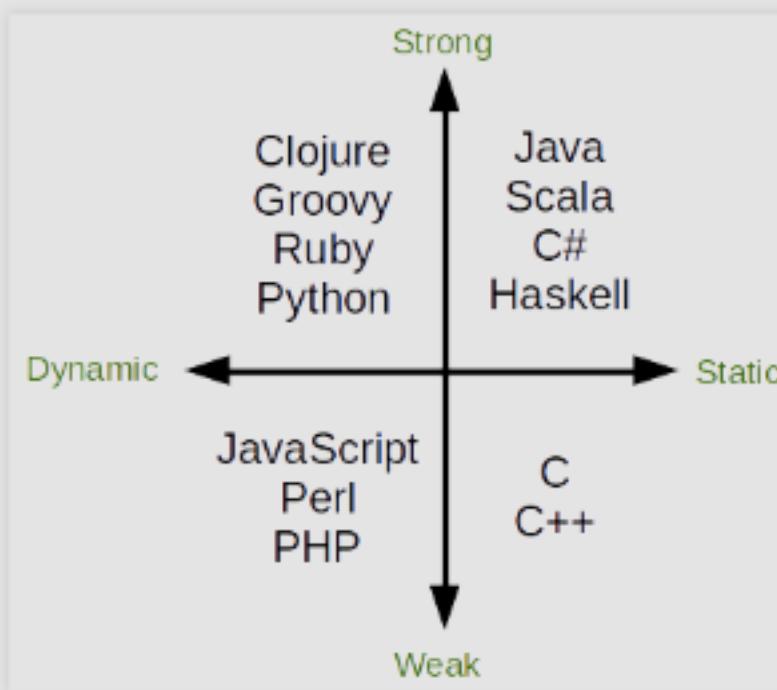
<https://github.com/tastejs/todomvc/>



LIKE JAVA BUT DIFFERENT



WEAK / DYNAMIC LANGUAGES



<http://nurkiewicz.github.io/talks/2015/haskell/>



DUCK TYPING

*(...) duck typing is a style of dynamic typing in which an object's **current set of methods and properties determines the valid semantics**, rather than its inheritance from a particular class or implementation of a specific interface.*

– Wikipedia

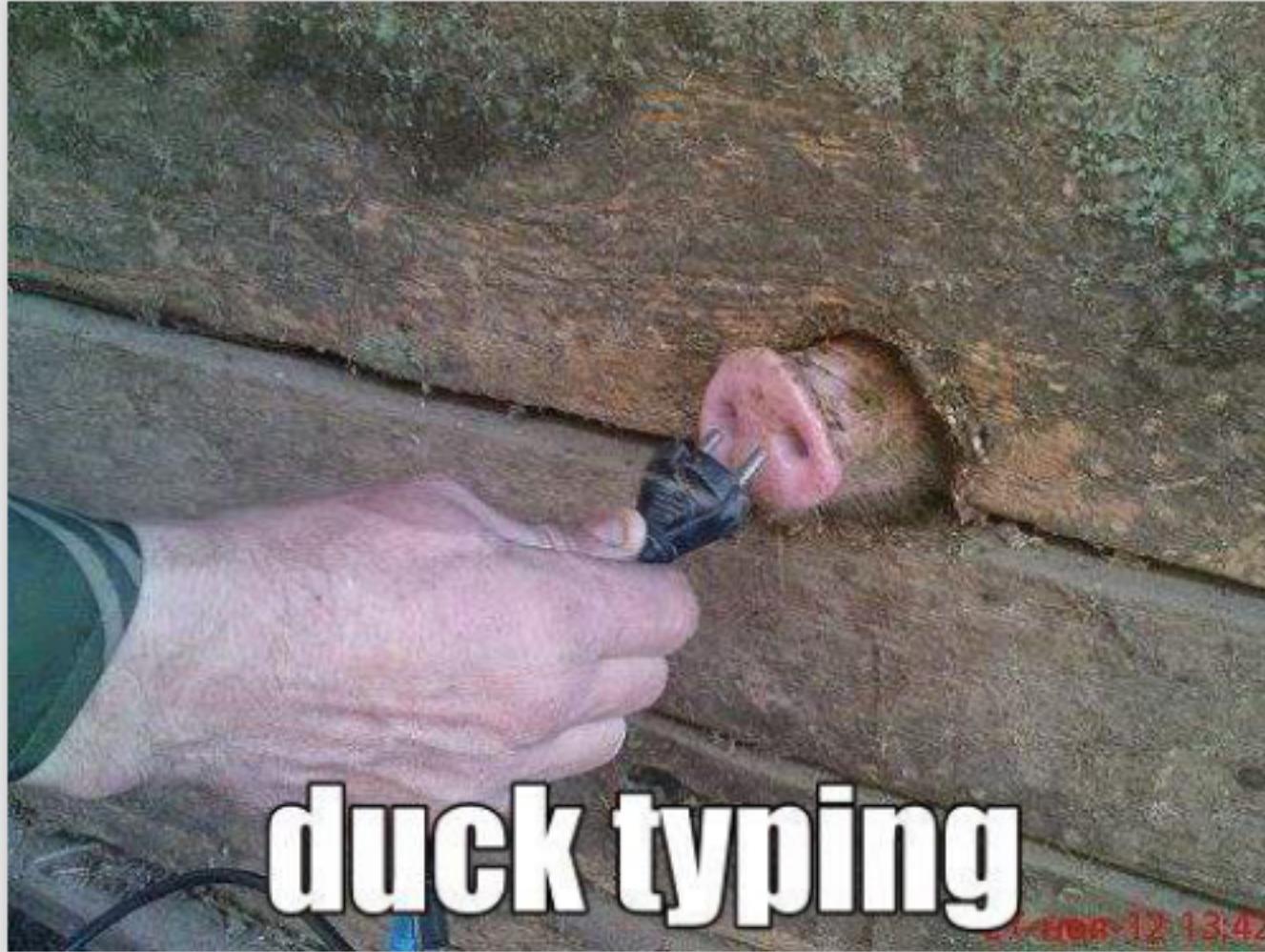


IN OTHER WORDS

*If it walks like a duck and it quacks like a duck,
then it is a duck.*

— Michael Youssef





THINGS TO WATCH OUT

- use strict
- var that = this
- operator new
- Module pattern



USE STRICT

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

<http://ejohn.org/blog/ecmascript-5-strict-mode-json-and-more/>

— John Resig



WHAT IS ECMASCIPT?

ECMAScript

standardized specification of JavaScript

ECMAScript 5

(2009) - to clarify ambiguities of all previous implementations

- Especially ECMAScript 3
- ECMAScript 4 was abandoned - politics

ECMAScript 6

(2015 - aka ECMAScript 2015) - since last year



VAR THAT = THIS

this in JavaScript

- function's this keyword behaves a little differently in JavaScript compared to other languages.
- it also has some differences between strict mode and non-strict mode.
- the value of this is determined by how a function is called.
- it may be different each time the function is called.



VAR THAT = THIS

```
var colours = ['red', 'green', 'blue'];
document.getElementById('element').addEventListener('click', function() {
    // this is a reference to the element clicked on

    var that = this;

    colours.forEach(function() {
        // this is undefined (attached to global)
        // that is a reference to the element clicked on
    });
});
```



VAR THAT = THIS

*By convention, we make a private **that** variable. This is used to make the object available to the private methods. This is a workaround for an error in the ECMAScript Language Specification which causes **this** to be set incorrectly for inner functions.*

<http://www.crockford.com/javascript/private.html>

— Douglas Crockford



OPERATOR NEW

- Functions in JavaScripts are objects
- new operator creates new object that inherits from Operand's prototype member
 - If new is forgotten - it's an ordinary function call
 - Meaning this refers to global windows not local function scope

```
jjs> var test = function() { print(this) }
jjs> test()
[object global]
jjs> new test()
[object Object]
```



OPERATOR NEW

- By convention - functions intended to be used with `new` should be given names with capital letters
- Convention gives visual clue that can spot some expensive mistakes

But it's even better not to use `new` at all



MODULE PATTERN

declares a function, which then calls itself immediately

```
var Module = (function () {  
    // code  
})();
```

- Think about it as Spring singleton beans



PRIVATE VS. PUBLIC

```
var Module = (function () {
    var _privateMethod = function () { // private
};

    var someMethod = function () { // public
};

    var anotherMethod = function () { // public
};

    return {
        someMethod: someMethod,
        anotherMethod: anotherMethod
    };
})();
```



INHERITANCE VS. COMPOSITION

```
var EnhancedModule = (function (Module) {  
    // access to 'Module'  
})(Module);
```



WHAT IS THE PROBLEM HERE?

- Stay close to 'bare metal'
- Play with simple tools (not over-abstract)
- Get **slapped** with all inconsistencies
 - The way here is the Crockford way - not the only way



*There are these two young fish swimming along,
and they happen to meet an older fish swimming
the other way, who nods at them and says,
"Morning, boys, how's the water?" And the two
young fish swim on for a bit, and then eventually
one of them looks over at the other and goes,
"What the hell is water?"*

<http://bulletin.kenyon.edu/x4280.html>

– David Foster Wallace



OMG! WE ARE DOOMED!



NO NEED TO WRITE VANILLA JS



NOT ONLY DOM FUNCTIONS

DOM manipulation

- jQuery
- Prototype
- React.js
- YUI Library

MVC (MVVM)

- AngularJS
- Backbone.js
- Chaplin.js
- Ember.js
- Knockout
- Meteor
- React.js



THE PARADOX OF CHOICE



BUT BEFORE WE START...



SEARCH DOCS...

QUICKSTART

TUTORIAL

BASICS

DEVELOPER GUIDE

COOKBOOK

API REFERENCE

Install these packages by opening a terminal window (command window in Windows) and running this npm command.

```
npm install
```



Scary **error messages in red** may appear during `install`. Ignore them. The install will succeed. See the [appendix below](#) for more information.

Our First Angular Component

The **Component** is the most fundamental of Angular concepts. A component manages a view - a piece of the web page where we display information to the user and respond to user feedback.

Technically, a component is a class that controls a view template. We'll write a lot of them as we build Angular apps. This is our first attempt so we'll keep it ridiculously simple.

Create an application source sub-folder

We like to keep our application code in a sub-folder off the root called `app/`. Execute the following command in the console window.

```
mkdir app  
cd app
```



[Tooling Integration](#)[Language Tooling](#)[Package Management](#)[Server-side Environments](#)[Add-Ons](#)[Animation](#)[Two-Way Binding Helpers](#)[Test Utilities](#)[Cloning Elements](#)[Keyed Fragments](#)[Immutability Helpers](#)[PureRenderMixin](#)[Performance Tools](#)[Shallow Compare](#)[Advanced Performance](#)[Context](#)

REFERENCE

[Top-Level API](#)[Component API](#)[Component Specs and Lifecycle](#)[Supported Tags and Attributes](#)[Event System](#)[DOM Differences](#)[Special Non-DOM Attributes](#)[Reconciliation](#)[Web Components](#)[React \(Virtual\) DOM Terminology](#)**Note:**

If you are using ES2015, you will want to also use the `babel-preset-es2015` package.

To install React DOM and build your bundle with browserify:

Code

```
$ npm install --save react react-dom babelify babel-preset-react  
$ browserify -t [ babelify ] main.js -o bundle.js
```

To install React DOM and build your bundle with webpack:

Code

```
$ npm install --save react react-dom babel-preset-react babel-loader babel-core  
$ webpack main.js bundle.js --module-bind 'js=babel-loader'
```

Note:

If you are using ES2015, you will want to also use the `babel-preset-es2015` package.

Note: by default, React will be in development mode, which is slower, and not advised for production. To use React in production mode, set the environment variable `NODE_ENV` to `production` (using `envify` or `webpack`'s `DefinePlugin`). For example:

Code

```
new webpack.DefinePlugin({  
  "process.env": {  
    NODE_ENV: JSON.stringify("production")  
  }  
});
```

[Underscore.js](#) (1.8.3)

- » GitHub Repository
- » Annotated Source
- » Underscore-contrib

Filter

[Introduction](#)

[Collections](#)

- each
- map
- reduce
- reduceRight
- find
- filter
- where
- findWhere
- reject
- every
- some
- contains
- invoke
- pluck
- max
- min
- sortBy
- groupBy
- indexBy
- countBy
- shuffle
- sample
- toArray
- size
- partition

[Arrays](#)

- first
- initial
- last
- rest
- compact

Installation

- **Node.js** `npm install underscore`
- **Meteor.js** `meteor add underscore`
- **Require.js** `require(["underscore"], ...)`
- **Bower** `bower install underscore`
- **Component** `component install jashkenas/underscore`



Collection Functions (Arrays or Objects)

each `_.each(list, iteratee, [context])` Alias: `forEach`

Iterates over a `list` of elements, yielding each in turn to an `iteratee` function. The `iteratee` is bound to the `context` object, if one is passed. Each invocation of `iteratee` is called with three arguments: `(element, index, list)`. If `list` is a JavaScript object, `iteratee`'s arguments will be `(value, key, list)`. Returns the `list` for chaining.

```
_.each([1, 2, 3], alert);
  => alerts each number in turn...
_.each({one: 1, two: 2, three: 3}, alert);
  => alerts each number value in turn...
```

Note: Collection functions work on arrays, objects, and array-like objects such as `arguments`, `NodeList` and similar. But it works by duck-typing, so avoid passing objects with a numeric `length` property. It's also good to note that an `each` loop cannot be broken out of — to break, use `_.find` instead.

map `_.map(list, iteratee, [context])` Alias: `collect`

Produces a new array of values by mapping each value in `list` through a transformation function (`iteratee`). The `iteratee` is passed three arguments: the `value`, then the `index` (or `key`) of the iteration, and finally a reference to the entire

SO THE BUILD TOOLS...



Joe Walnes

@joewalnes

If you only need to target modern browsers,
one of the most productive things you can do
is eliminate JS build tools from your workflow

RETWEETS
29

LIKES
57



1:34 AM - 3 May 2016

<https://twitter.com/joewalnes/status/727280111511195648>



... AS COMPLEXITY SELLS BETTER

“Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.”

—**Edsger Dijkstra**

*Source/Notes:

Source: EWD896 - On the nature of Computing Science

<https://twitter.com/denormalize/status/679399609991430148/>



... WE WON'T USE THEM



Joe Walnes
@joewalnes

No WebPack, no Babel... Just good old <script src=...> pointing to your code. And a text editor. No distractions and rabbit holes of doom.

RETWEETS LIKES
13 27



1:37 AM - 3 May 2016

<https://twitter.com/joewalnes/status/727280857476538368>



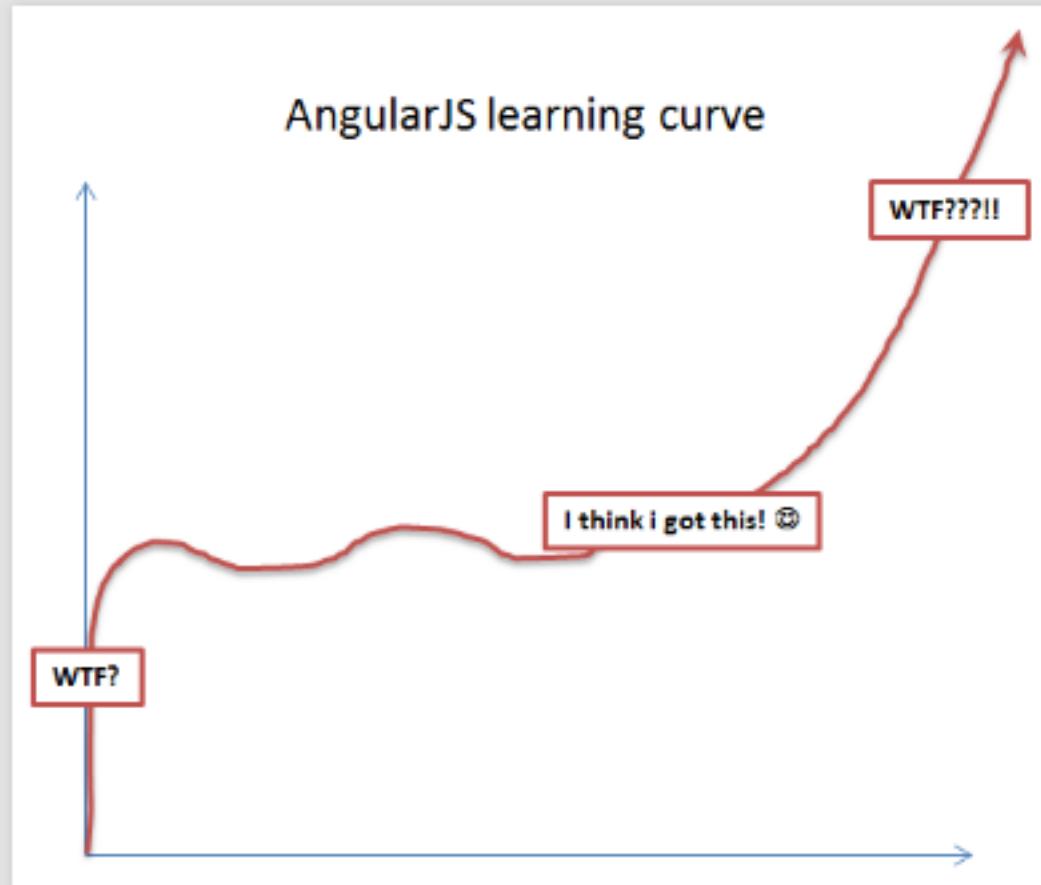
LEARNING PROCESS

- We should expect tool to be working straightforwardly
- We don't need all this hot reloading, file watching, ES2015 at the beginning
- Start with writing The Application
 - Add in the developer functionality when felt more comfortable in the environment

Learn just one thing at a time



LEARNING CURVE



WE WANT IT TO BE SIMPLE

```
<div id='root'></div>

<script src="https://fb.me/react-15.0.1.js"></script>
<script src="https://fb.me/react-dom-15.0.1.js"></script>

<script>
  var HelloMessage = React.createClass({
    render: function () {
      return React.DOM.h1({className: 'title'}, 'Hello ${this.props.name}');
    }
  });

  var h1 = React.createFactory(HelloMessage);
  ReactDOM.render(h1({name: 'Jakub'}), document.getElementById('root'));
</script>
```



WHAT ABOUT THE JS ITSELF?



As we embrace in JavaScript



As we embrace in JavaScript

- No typing



As we embrace in JavaScript

- No typing
- No modules



As we embrace in JavaScript

- No typing
- No modules
- Verbose patterns (*or call them workarounds*)



THE TYPESCRIPT'S PROMISE

- A superset of JavaScript
- Optionally typed (with modules, namespaces)
- Compiles to ECMAScript3 (backwards compatibility)
- No special runtime
- Aligned with ES6 features



Microsoft's TypeScript may be the best of the many JavaScript front ends. It seems to generate the most attractive code. And I think it should take pressure off of the ECMAScript Standard for new features like type declarations and classes. Anders has shown that these can be provided nicely by a preprocessor, so there is no need to change the underlying language.

<https://plus.google.com/u/0/+DouglasCrockfordEsq/posts/MgzNUSTwjRt>

— Douglas Crockford



I think that JavaScript's loose typing is one of its best features and that type checking is way overrated. TypeScript adds sweetness, but at a price. It is not a price I am willing to pay.

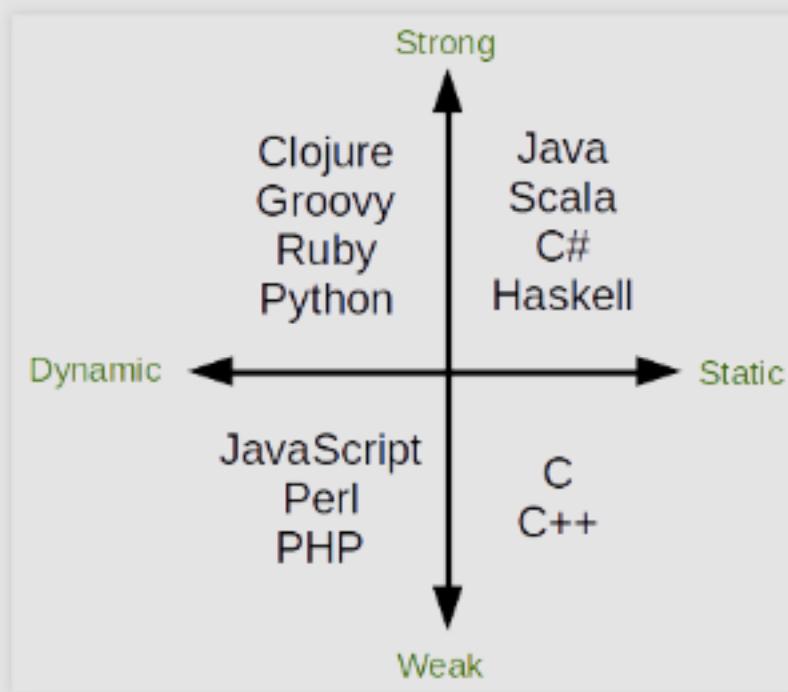
<https://plus.google.com/u/0/+DouglasCrockfordEsq/posts/MgzNUSTwjRt>

— Douglas Crockford



A STEP BACK

Weak / dynamic languages



<http://nurkiewicz.github.io/talks/2015/haskell/>



SOME EXAMPLES

<http://www.typescriptlang.org/play/index.html>



SO WHY TYPESCRIPT



SO WHY TYPESCRIPT

- Scalable HTML5 client-side development



SO WHY TYPESCRIPT

- Scalable HTML5 client-side development
- Modular development



SO WHY TYPESCRIPT

- Scalable HTML5 client-side development
- Modular development
- Easily learnable for Java developers



SO WHY TYPESCRIPT

- Scalable HTML5 client-side development
- Modular development
- Easily learnable for Java developers
- Non-invasive (existing libs, browser support)



SO WHY TYPESCRIPT

- Scalable HTML5 client-side development
- Modular development
- Easily learnable for Java developers
- Non-invasive (existing libs, browser support)
- Long-term vision



SO WHY TYPESCRIPT

- Scalable HTML5 client-side development
- Modular development
- Easily learnable for Java developers
- Non-invasive (existing libs, browser support)
- Long-term vision
- Clean JS output (exit strategy)



BUILD TOOLS NOT REQUIRED

```
<script  
  src="https://rawgit.com/Microsoft/TypeScript/master/lib/typescriptServices.js">  
</script>  
<script  
  src="https://rawgit.com/basarat/typescript-script/master/transpiler.js">  
</script>
```



RUN IT IN A BROWSER

```
<script type="text/typescript">
  module Sayings {
    export class Greeter {
      greeting: string;
      constructor (message: string) {
        this.greeting = message;
      }
      greet() {
        return "Hello, " + this.greeting;
      }
    }
  }
</script>
```



RUN IT IN A BROWSER

```
<script type="text/typescript">
    var greeter = new Sayings.Greeter('world');
    var button = document.createElement('button')
    button.innerHTML = "Say hello"
    button.onclick = function() {
        alert(greeter.greet())
    }
    document.body.appendChild(button)
</script>
```



ANY ALTERNATIVES



ANY ALTERNATIVES

- CoffeeScript



ANY ALTERNATIVES

- CoffeeScript
- ECMAScript6 (ES2015) - with babel



ANY ALTERNATIVES

- CoffeeScript
- ECMAScript6 (ES2015) - with babel
- ScalaJS



ANY ALTERNATIVES

- CoffeeScript
- ECMAScript6 (ES2015) - with babel
- ScalaJS
- Kotlin



ANY ALTERNATIVES

- CoffeeScript
- ECMAScript6 (ES2015) - with babel
- ScalaJS
- Kotlin
- clojurescript



SHIMS AND POLYFILL

Polyfill

retrofits legacy browsers with modern HTML5/CSS3 features

Shims

performs interception of an API call and provides a layer of abstraction



BABEL

Babel has support for the latest version of JavaScript through syntax transformers. These plugins allow you to use new syntax, right now without waiting for browser support.



SCALAJS

```
object ScalaJSExample extends js.JSApp {  
  
    def main() = {  
        var x = 0  
        while(x < 10) {  
            println(x)  
            x += 3  
        }  
        println(s"Final: $x")  
  
    }  
}
```



SCALAJS - TRANSPILED

```
ScalaJS.c.LMain$.prototype.main__V = (function() {
    var x = 0;
    while ((x < 10)) {
        ScalaJS.m.s_Predef$()
            .println__0__V(x)
        x = ((x + 3) | 0)
    };
    ScalaJS.m.s_Predef$()
        .println__0__V("Final: " + x)
});
```



TYPESCRIPT

```
class Main {  
    hello() {  
        var x = 0;  
        while (x < 10) {  
            console.log(x)  
            x += 3  
        }  
  
        console.log(`Final: ${x}`)  
    }  
}  
  
var main = new Main()  
main.hello()
```



TYPESCRIPT - TRANSPILED

```
var Main = (function () {
    function Main() {
    }
    Main.prototype.hello = function () {
        var x = 0;
        while (x < 10) {
            console.log(x);
            x += 3;
        }
        console.log("Final: " + x);
    };
    return Main;
})();

var main = new Main();
main.hello();
```



TREAT IT AS A GOOD CONVENTION



TREAT IT AS A GOOD CONVENTION

- Always look on the output of the transpiler



TREAT IT AS A GOOD CONVENTION

- Always look on the output of the transpiler
- Learn how proper JavaScript may look like



TREAT IT AS A GOOD CONVENTION

- Always look on the output of the transpiler
- Learn how proper JavaScript may look like
- Remove TypeScript and just JavaScript



TREAT IT AS A GOOD CONVENTION

- Always look on the output of the transpiler
- Learn how proper JavaScript may look like
- Remove TypeScript and just JavaScript
- Without the Bad Parts



... WE USED

- Browser
- <script /> tag
- React.js
- Typescript

Because it is an **opinionated talk**



WHY SHOULD BE EVEN BOTHER?



WHY SHOULD BE EVEN BOTHER?

- There is not such thing as full-stack developer



WHY SHOULD BE EVEN BOTHER?

- There is not such thing as full-stack developer
- It's a **unicorn**



WHY SHOULD BE EVEN BOTHER?

- There is not such thing as full-stack developer
- It's a **unicorn**
- But there is a full-stack architect



WHY SHOULD BE EVEN BOTHER?

- There is not such thing as full-stack developer
- It's a **unicorn**
- But there is a full-stack architect
- A person who feels appropriately comfortable in both sides



WHY SHOULD BE EVEN BOTHER?

- There is not such thing as full-stack developer
- It's a **unicorn**
- But there is a full-stack architect
- A person who feels appropriately comfortable in both sides
- Be a **full-stack architect**



QUESTIONS?

