HTML5 Web Component Fundamentals

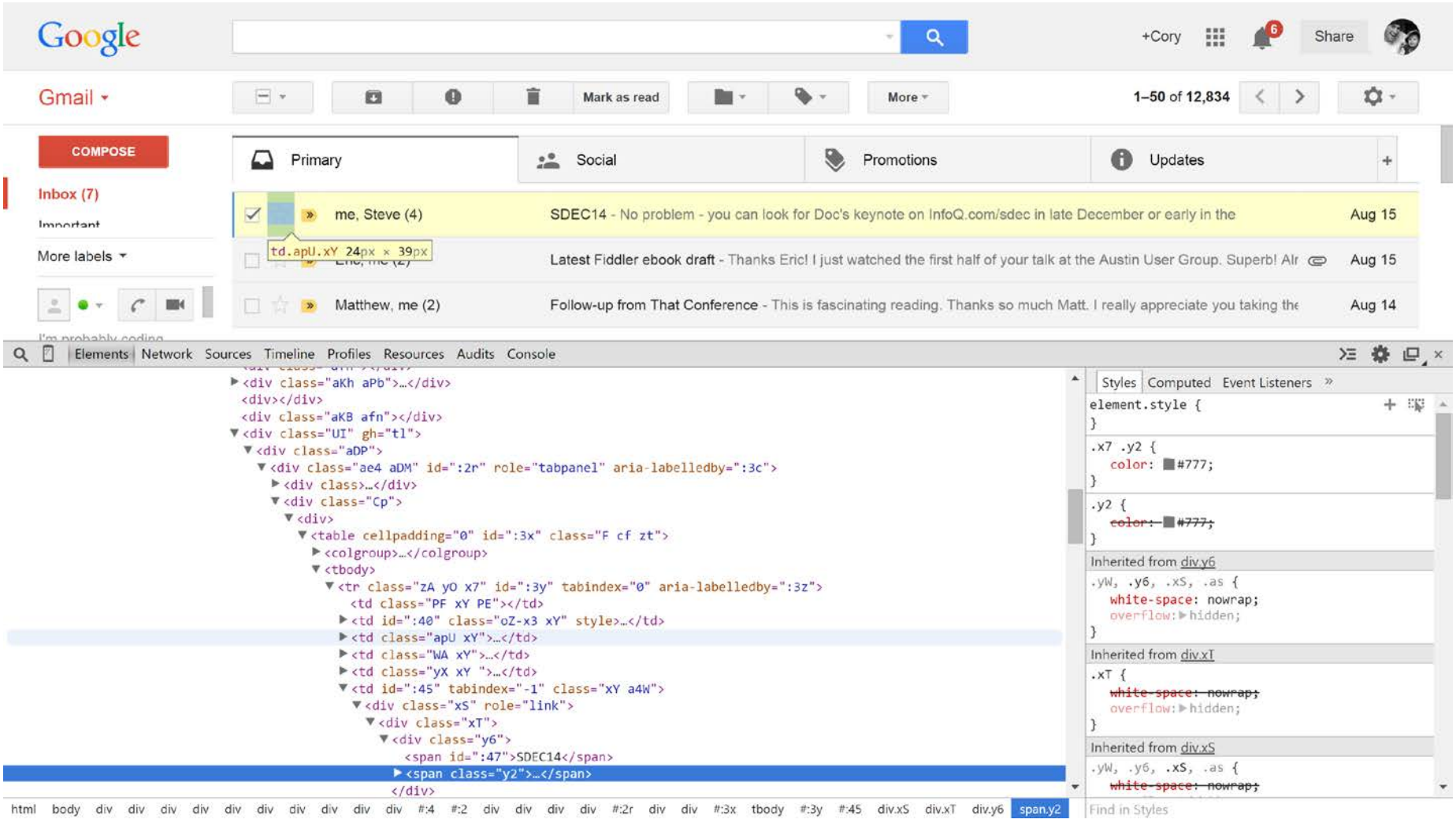
Introduction

Cory House
bitnative.com
Twitter: @housecor

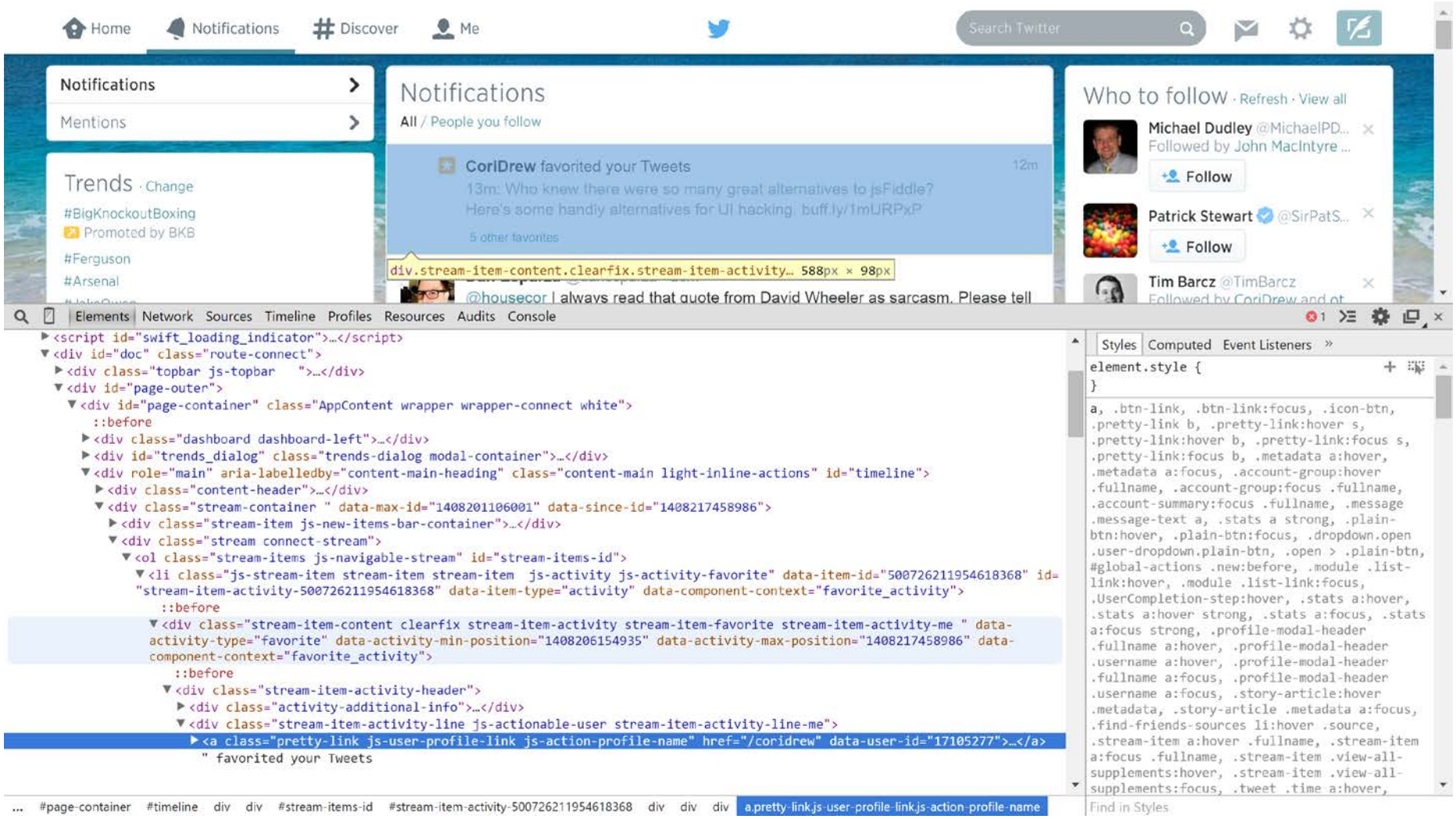
pluralsight
hardcore developer training

Web Developers Have Problems

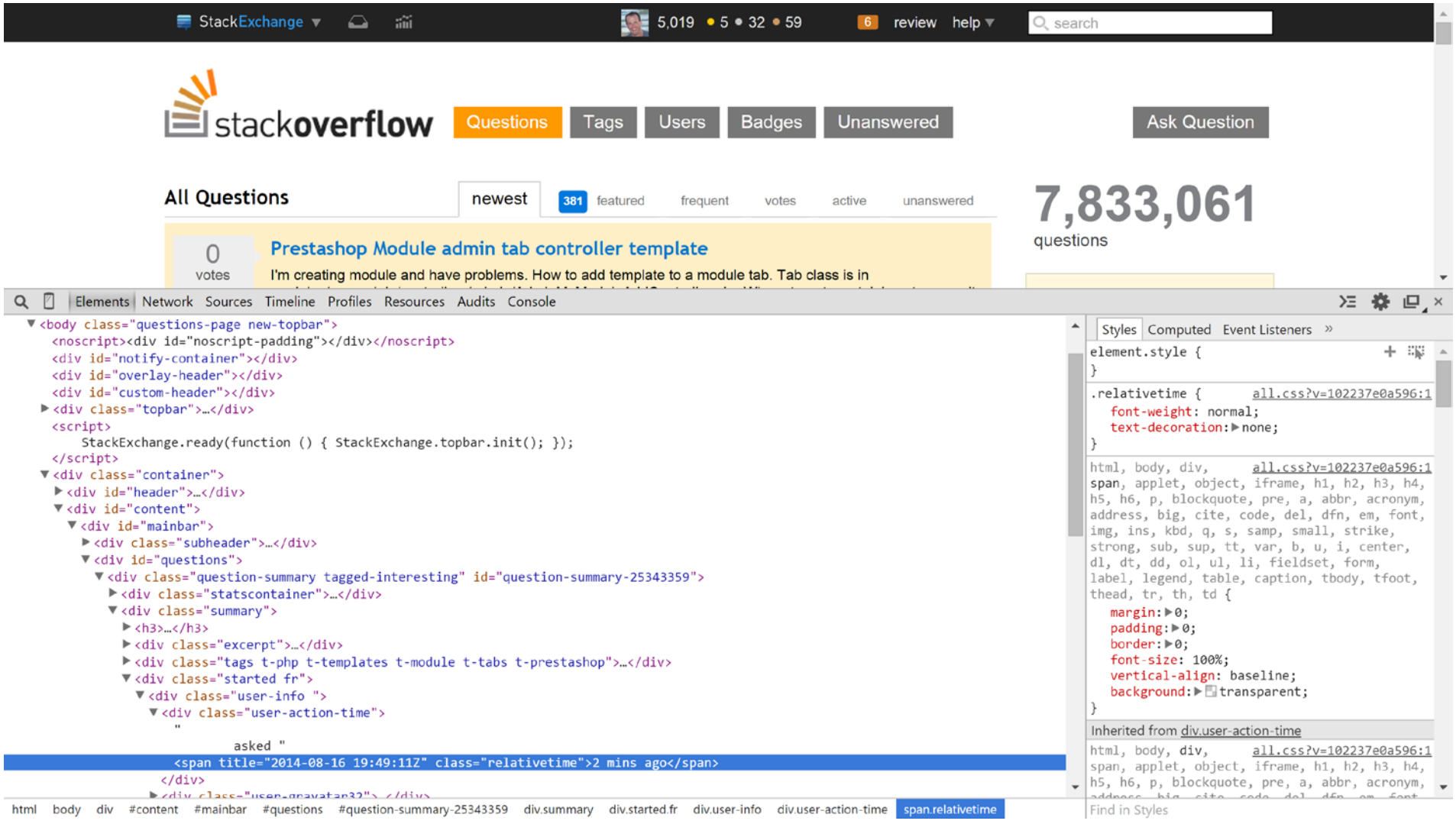
Problem #1: Undescriptive Markup



Problem #1: Undescriptive Markup



Problem #1: Undescriptive Markup



Problem #2: Style Conflicts

Avoiding style conflicts requires highly specific CSS selectors

Use of !important to force styles

No guarantee another style won't conflict

We envision this

But may end up with this

Problem #3: No Native Templates

We slap HTML in `<script type="text/html">`

We store HTML in hidden DOM elements and manually extract

We use iframe's to get separate scope and styling

Problem #4: No Bundling

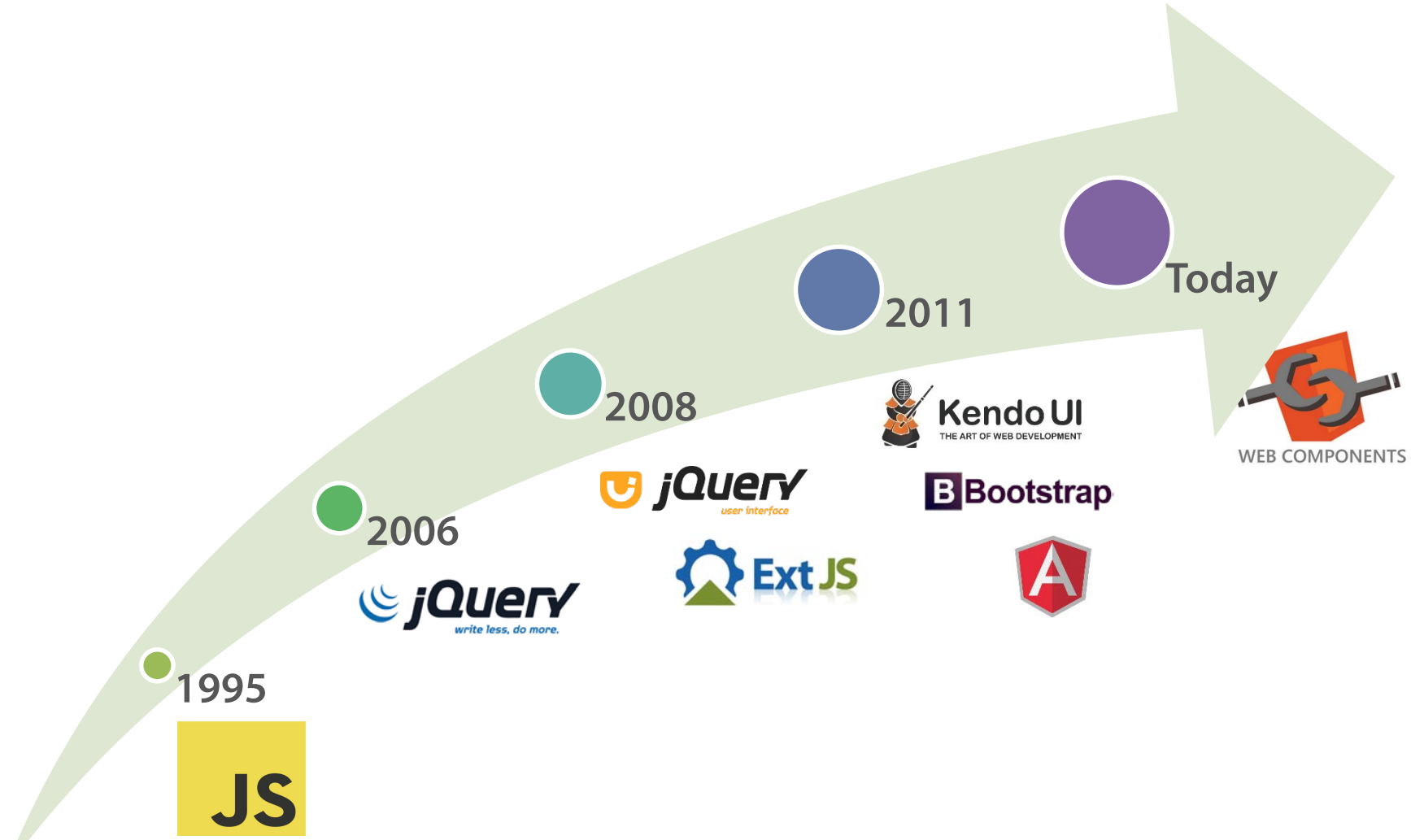
What if you want to bundle some CSS, JS, and HTML together?

Consider Bootstrap…

```
<script src="jquery.min.js"></script>
<script src="bootstrap.min.js"></script>
<link rel="stylesheet" href="bootstrap.min.css" />
<script src="script.js"></script>
<link rel="stylesheet" href="style.css" />
```

Problem #5: No Standard

A Path to Standard Components



5 Problems

1. Undescriptive markup
2. No component standard
3. Style conflicts
4. No native templates
5. No bundling

The Solution?

WEB COMPONENTS

Templates

Inert, reusable markup

Module 2

Custom Elements

Define our own Elements

Module 3

Shadow DOM

Encapsulated Markup & Styling

Modules 4 - 6

Imports

Bundle HTML, JS, & CSS

Module 7

Prefer demos
that actually work?

Then use this.

Are We Componentized Yet?

Tracking the progress of Web Components through standardisation, polyfillification[1], and implementation.

	Documentation		Implementation				
	Explained	Specced	Polyfill	Chrome / Opera	Firefox	Safari	IE
Templates				Stable	Stable	8	
HTML Imports				Stable	Flag		
Custom Elements				Stable	Flag		
Shadow DOM				Stable	Flag		

http://jonrimmer.github.io/

Why Learn Native Web Components First?

Learn this

Foundation

Before you learn this

Abstraction over foundation

Learn this

Foundation

Before you learn this

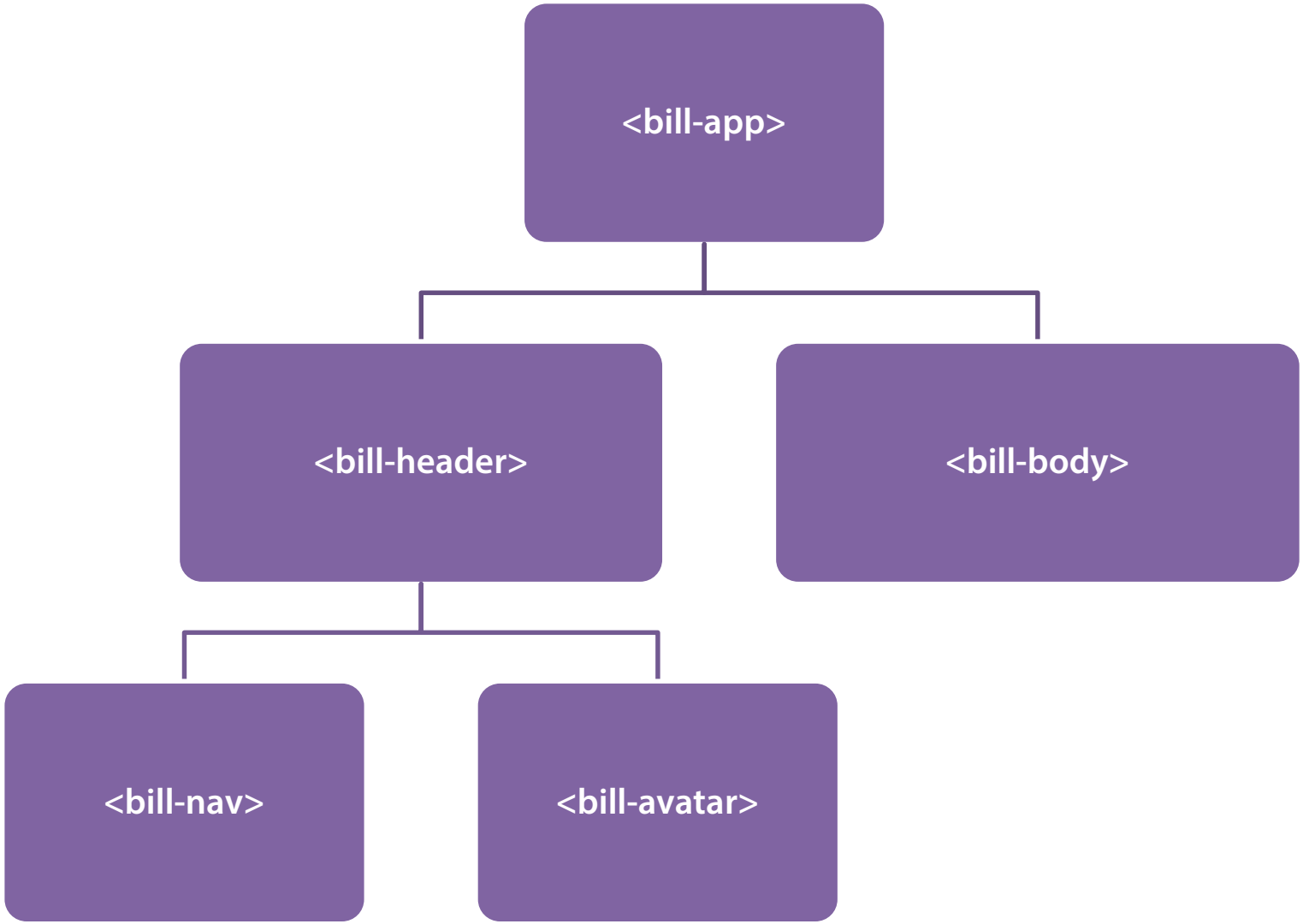
Abstraction over foundation

What Level of Abstraction?

Any.

Low Level or Entire App!

What Level of Abstraction?



Cory House

Cory is an independent consultant, INETA speaker, Microsoft MVP, blogger, and software architect with 15 years of experience in software development. As a software architect at VinSolutions, Cory specializes in creating C#, .Net and JavaScript based single page applications for the automotive industry. He regularly speaks on clean code, architecture, and software career development at conferences and user groups across the country. Cory blogs at bitnative.com and is active on Twitter as @housecor.

MVP Microsoft Most Valuable Professional

Courses by Cory

Architecting Applications for the Real World in .NET
by Cory House — Intermediate — 2h 52m — 07 Jan 2014

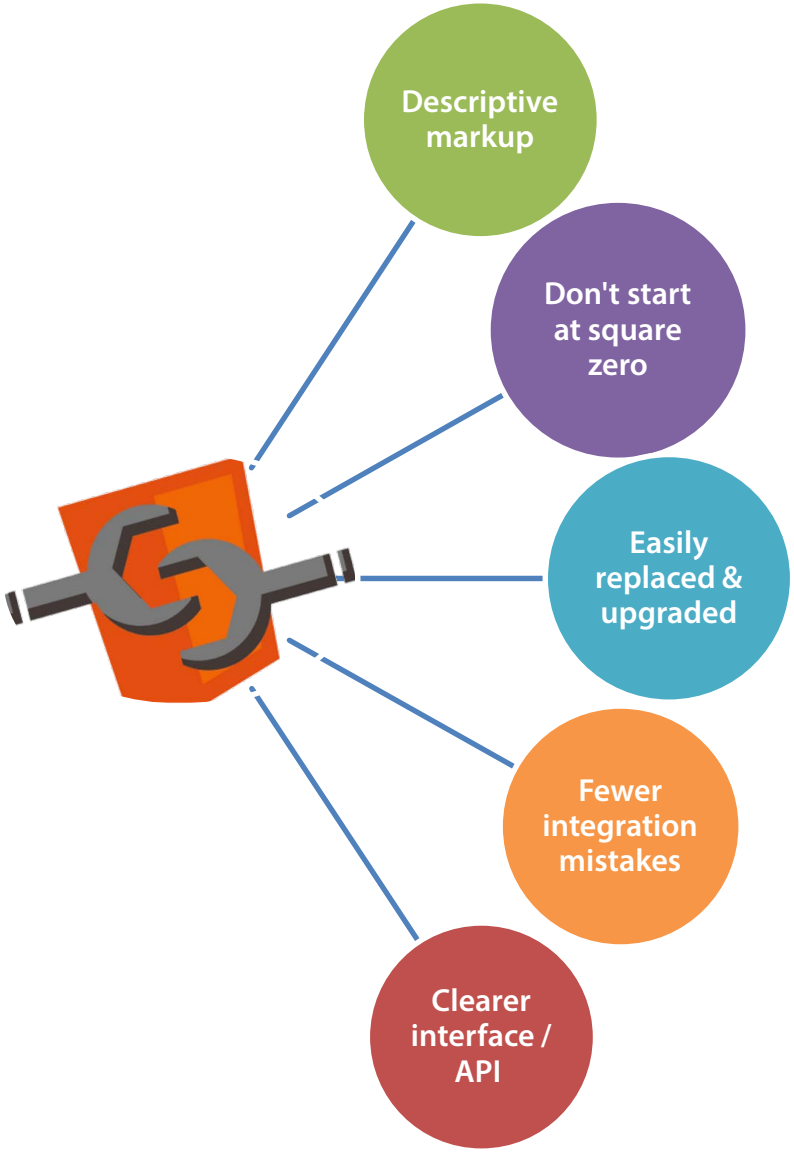
Becoming an Outlier: Reprogramming the Developer Mind
by Cory House — Intermediate — 2h 33m — 24 Apr 2014

Clean Code: Writing Code for Humans
by Cory House — Intermediate — 3h 10m — 07 Oct 2013

The Dawn of the Democratic Web

Communities will rally around components

Github will measure interest

Popular components will drive future HTML elements

Why Web Components?