

The Modern Java Web Developer

Matt Raible • <http://raibledesigns.com>



Who is Matt Raible?

Father, Skier, Cyclist

Web Framework Connoisseur

Founder of [AppFuse](#)

Blogger on [raibledesigns.com](#)

How about YOU?

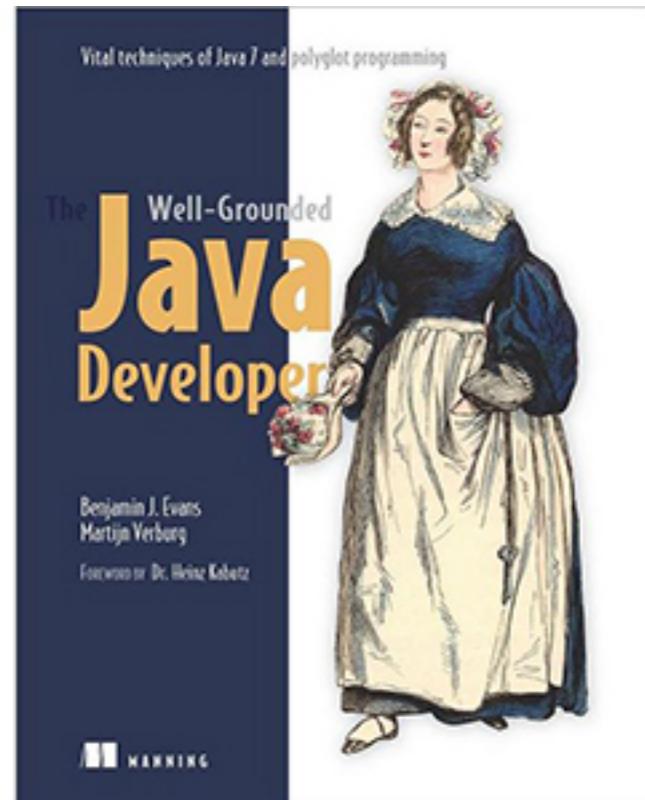
- Have you developed a Struts 1 application? Used PHP?
- Have you every written CSS from scratch?
- Why do you hate JavaScript?
- What's your favorite JavaScript framework?
- What do you want to learn from this talk?



by Art Escobedo

Topic Inspiration

- Inspired by Ben Evans's and Martijn Verburg's **The Well-Grounded Java Developer**
 - Developing with Java 7
 - Vital techniques
 - Polyglot programming on the JVM
 - Crafting the polyglot project



Purpose



The Modern Java Web Developer



The Modern **JVM** Web Developer

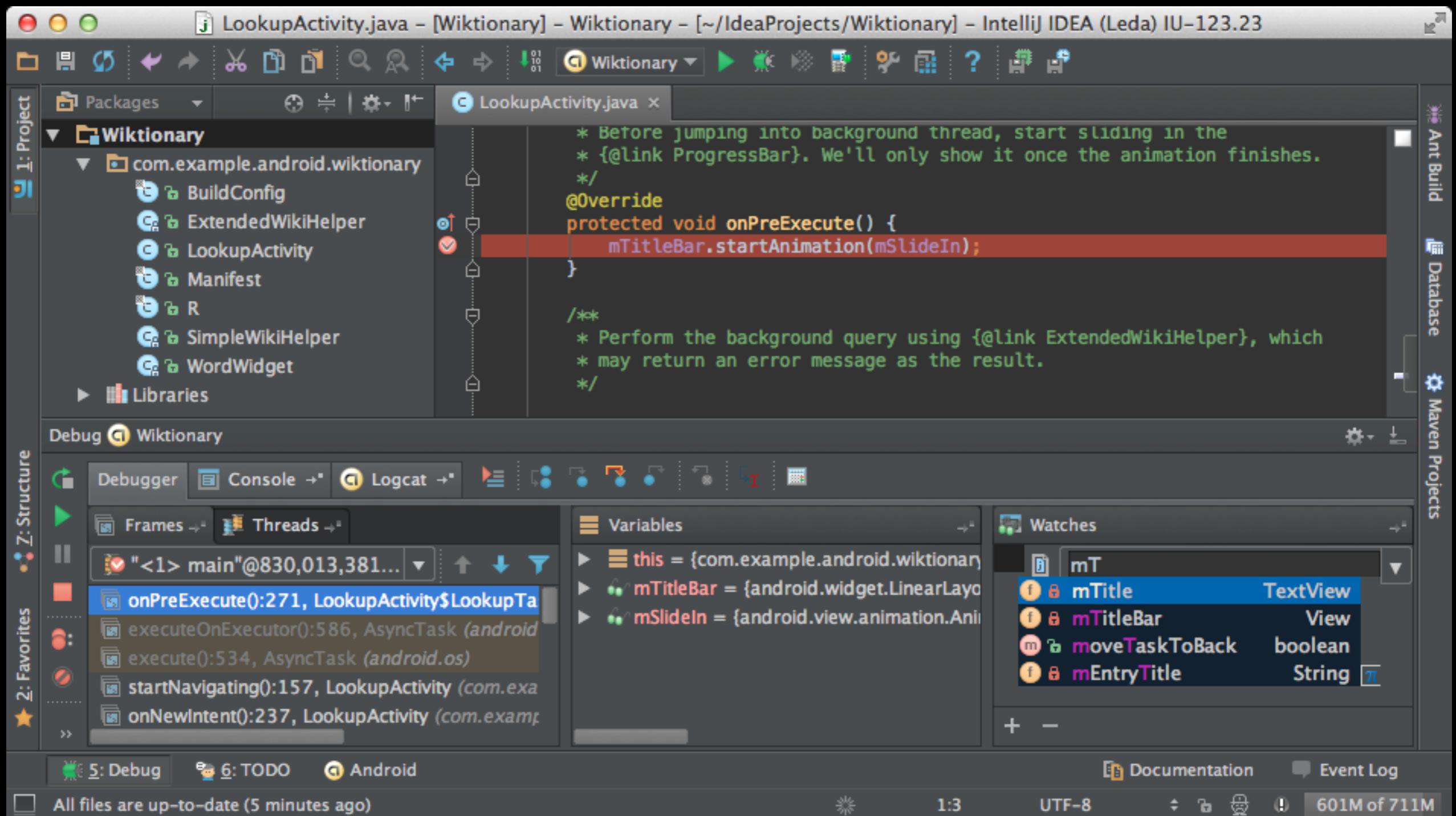
- Starts with Fast Hardware
- Uses IntelliJ IDEA
- Leverages jQuery, HTML5, and CSS3
- Creates High Performance Web Sites
- For Mobile Devices
- In the Cloud
- And *cares* about Security



Fast Hardware

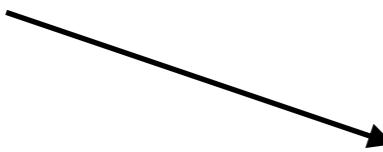


IntelliJ



Supports Zen Coding

```
div#page>div.logo+ul#navigation>li*5>a
```



```
<div id=page>
  <div class=logo></div>
  <ul id=navigation>
    <li><a href=></a></li>
    <li><a href=></a></li>
    <li><a href=></a></li>
    <li><a href=></a></li>
    <li><a href=></a></li>
  </ul>
</div>
```

Java 7 and 8

- Strings in switch statements
- Improved exception handling with multi-catch
- Try with resources
- Diamond Syntax
- NIO.2: Path, Files and Asynchronous I/O



```
Path path = FileSystems.getDefault().getPath(logs, access.log);
BufferedReader reader = Files.newBufferedReader(path, StandardCharsets.UTF_8);
```

Java 7 and 8

- Functional Interfaces with default method
- Lambda Expressions (a.k.a. Closures)

```
// sort a list by lastName
List<Person> persons = ...;
persons.sort((p1, p2) ->
    p1.getLastName().compareTo(p2.getLastName()));
```

- JSR 310 Date and Time API
- Parallel Collections
- Nashorn JavaScript Engine



The Modern JVM Web Developer is aware of...

- %s/Hibernate/Spring Data
- Servlet 3
 - @WebServlet, @WebFilter, @WebListener
 - Asynchronous Servlets
 - WebApplicationInitializer (to eliminate web.xml)
- Functional Programming
- REST



Scala

Scala is like the dragon in Avatar. It will try to kill you, but if you master it, you can fly great distances with it and have a wonderful time.

-- Venkat Subramaniam



Scala Basics

- `def` starts a method
- variables are started with `var` or `val`
- variables are defined with name:type
- semicolons are not required

```
import play.mvc.Http

trait Scalate {

    def render(args: (Symbol, Any)*) = {
        val template = Http.Request.current().action.replace(".", "/")
        ScalateTemplate(template).render(args: _*)
    }
}
```

Scala vs. Java

```
public class Car {  
    private final int year;  
    private int miles;  
  
    public int getYear() { return year; }  
    public int getMiles() { return miles; }  
    public void setMiles(int theMiles) { miles = theMiles; }  
  
    public Car(int theYear, int theMiles) {  
        year = theYear;  
        miles = theMiles;  
    }  
}
```



Scala vs. Java

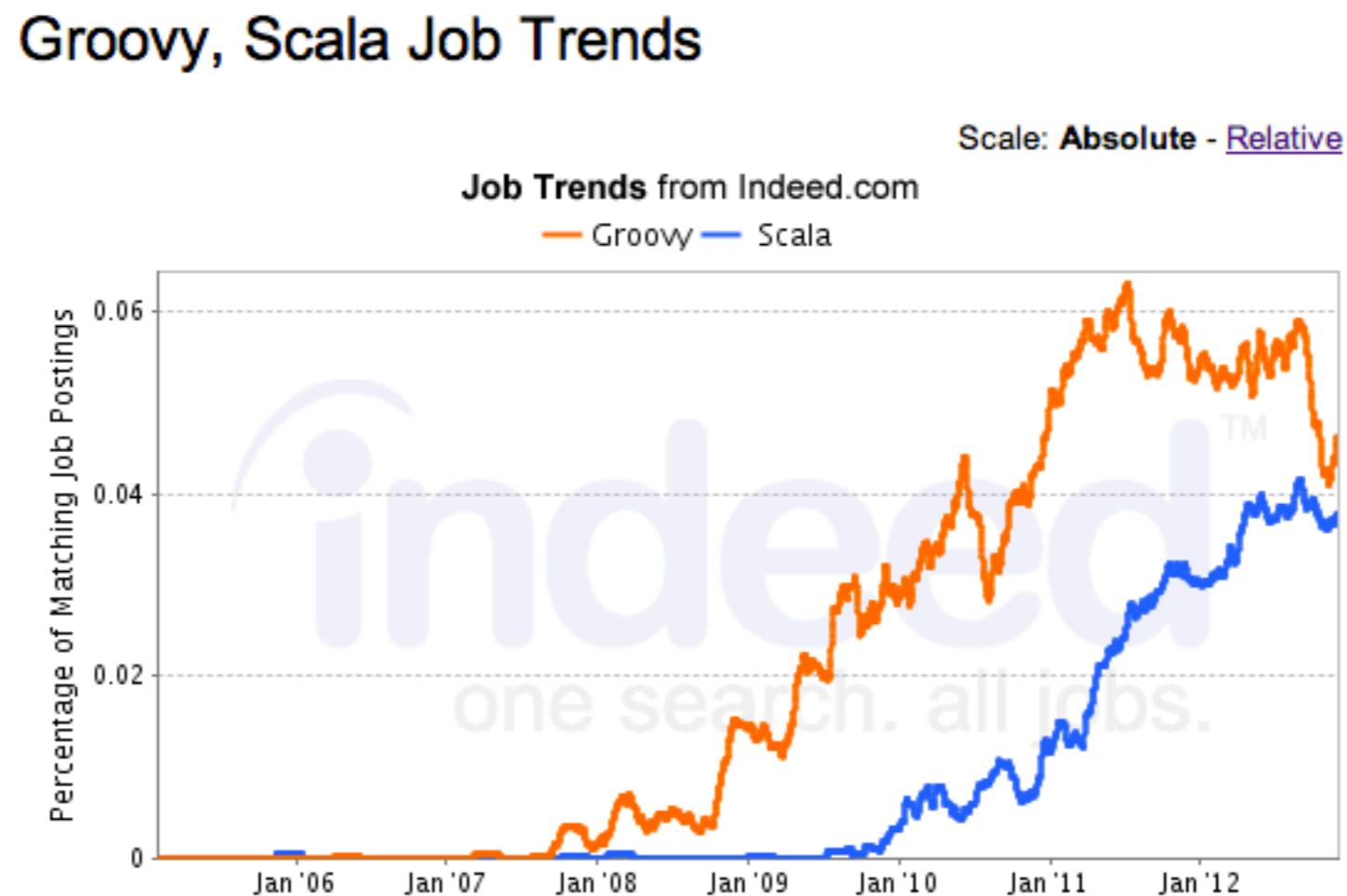
```
class Car(val year: Int, var miles: Int)
```



What about Groovy?



Groovy is still hot...



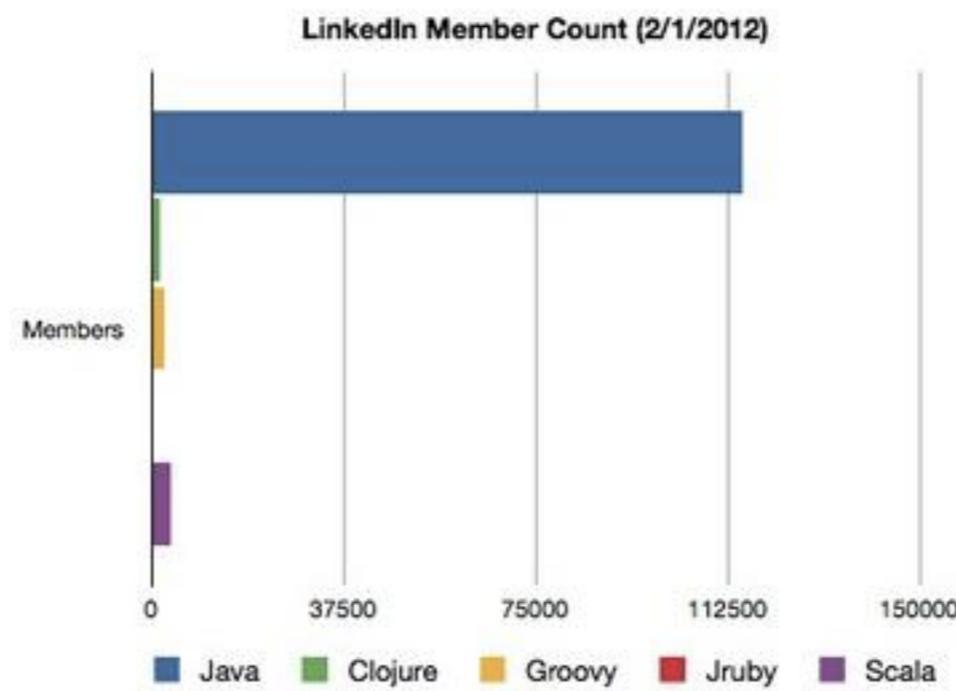
Learning Scala

- **Scala for the Impatient** - Cay Horstmann
- **Programming in Scala, 2nd Edition** - Martin Odersky, Lex Spoon, and Bill Venners
- Functional Programming Principles in Scala
 - Mar 25th 2013 (7 weeks long)
 - 5-7 Hours per week
- Signup at <https://www.coursera.org/course/progfun>



The Java Language

Java remains – in spite of the fragmented programming language landscape – a viable, growing language.



<http://redmonk.com/sogrady/2012/02/08/language-rankings-2-2012/>

and ... it's still the most popular!

FEBRUARY 08, 2013

Java retakes the lead in language popularity

Recent security problems aside, Java moves past C in Tiobe's language popularity index thanks to its ties to Android development

By [Paul Krill](#) | [InfoWorld](#)

[Follow @pjkrill](#)



[Print](#)

| [27 Comments](#)



[Twitter](#)



[StumbleUpon](#)



[LinkedIn](#)



[Google+](#)



[Reddit](#)



[Facebook](#)

Like

77



[Email](#)



[Link](#)



[More](#)

Despite recent headlines about its security woes, Java has returned to the top spot in a monthly assessment of the popularity of programming languages, [10 months after being supplanted by the C language](#).

The Tiobe Programming Community Index again has Java as the most popular language. The company says Java's ties to Android mobile development are the reason for its recent good fortune. "Boosted by the success of Android phones, Java has gained most market share of all languages last half a year," Tiobe said.



The Services Developer

The screenshot shows a web browser displaying the InfoQ website at www.infoq.com/articles/webber-rest-workflow. The page title is "InfoQ: How to GET a Cup of Coffee". The main content area features the article title, author information (Jim Webber, Savas Parastatidis & Ian Robinson, Oct 02, 2008), and a summary of the article's purpose. The article discusses how to use the Web as a primary design philosophy for integration scenarios. The page includes navigation links for various software development topics like Development, Architecture & Design, Process & Practices, Operations & Infrastructure, and Enterprise Architecture. A sidebar on the right promotes QCon conferences in London and New York. The bottom of the page contains a "Bookmark this!" button and social sharing icons.

InfoQ: How to GET a Cup of Coffee

Posted by Jim Webber, Savas Parastatidis & Ian Robinson on Oct 02, 2008

Sections Enterprise Architecture, Architecture & Design Topics Workflow / BPM, Business Process Management, **FEATURED SOA**, REST, Business, Enterprise Architecture, Web services, Architecture, Design Guideline, Patterns

Share

We are used to building distributed systems on top of large middleware platforms like those implementing CORBA, the Web Services protocols stack, J2EE, etc. In this article, we take a different approach, treating the protocols and document formats that make the Web tick as an application platform, which can be accessed through lightweight middleware. We showcase the role of the Web in application integration scenarios through a simple customer-service interaction scenario. In this article, we use the Web as our primary design philosophy to distil and share some of the thinking in our forthcoming book "GET /connected - Web-based integration" (working title).

Introduction

The integration domain as we know it is changing. The influence of the Web and the trend towards more agile practices are challenging our notions of what constitutes good integration. Instead of being a specialist activity conducted in the void between systems – or even worse, an afterthought – integration is now an everyday part of successful solutions.

Yet, the impact of the Web is still widely misunderstood and underestimated in enterprise computing. Even those who are Web-savvy often struggle to understand that the Web isn't about middleware solutions supporting XML over HTTP, nor is it a crude RPC mechanism. This is a shame because the Web has much more value than simple point-to-point connectivity; it is in fact a robust integration platform.

In this article we'll showcase some interesting uses of the Web, treating it as a pliant and robust platform for doing very cool things with enterprise systems. And there is nothing that typifies enterprise software more than workflows...

Fast APIs



JAX-RS

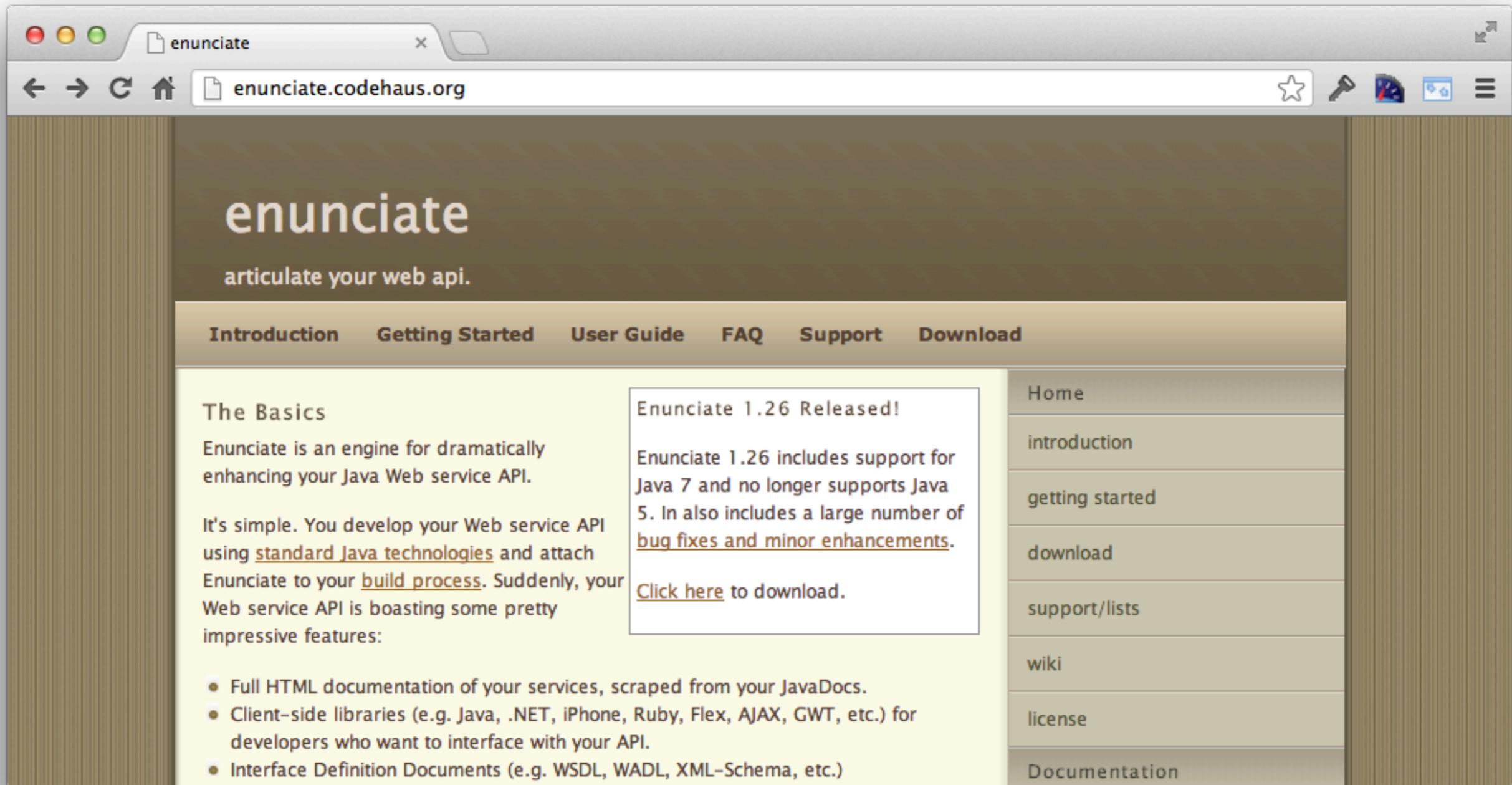


Dropwizard



AppFuse

Document Your API



The screenshot shows a web browser window with the title bar "enunciate" and the URL "enunciate.codehaus.org". The page itself has a brown header with the word "enunciate" and the tagline "articulate your web api.". Below the header is a navigation menu with links to "Introduction", "Getting Started", "User Guide", "FAQ", "Support", and "Download". The main content area contains several sections: "The Basics" (describing Enunciate as an engine for enhancing Java Web service APIs), "Enunciate 1.26 Released!" (announcing support for Java 7 and bug fixes), and a bulleted list of features. To the right is a sidebar with links to "Home", "introduction", "getting started", "download", "support/lists", "wiki", "license", and "Documentation".

enunciate
articulate your web api.

Introduction Getting Started User Guide FAQ Support Download

The Basics
Enunciate is an engine for dramatically enhancing your Java Web service API. It's simple. You develop your Web service API using [standard Java technologies](#) and attach Enunciate to your [build process](#). Suddenly, your Web service API is boasting some pretty impressive features:

- Full HTML documentation of your services, scraped from your JavaDocs.
- Client-side libraries (e.g. Java, .NET, iPhone, Ruby, Flex, AJAX, GWT, etc.) for developers who want to interface with your API.
- Interface Definition Documents (e.g. WSDL, WADL, XML-Schema, etc.)

Enunciate 1.26 Released!
Enunciate 1.26 includes support for Java 7 and no longer supports Java 5. It also includes a large number of [bug fixes and minor enhancements](#). [Click here to download.](#)

Home
[introduction](#)
[getting started](#)
[download](#)
[support/lists](#)
[wiki](#)
[license](#)
Documentation

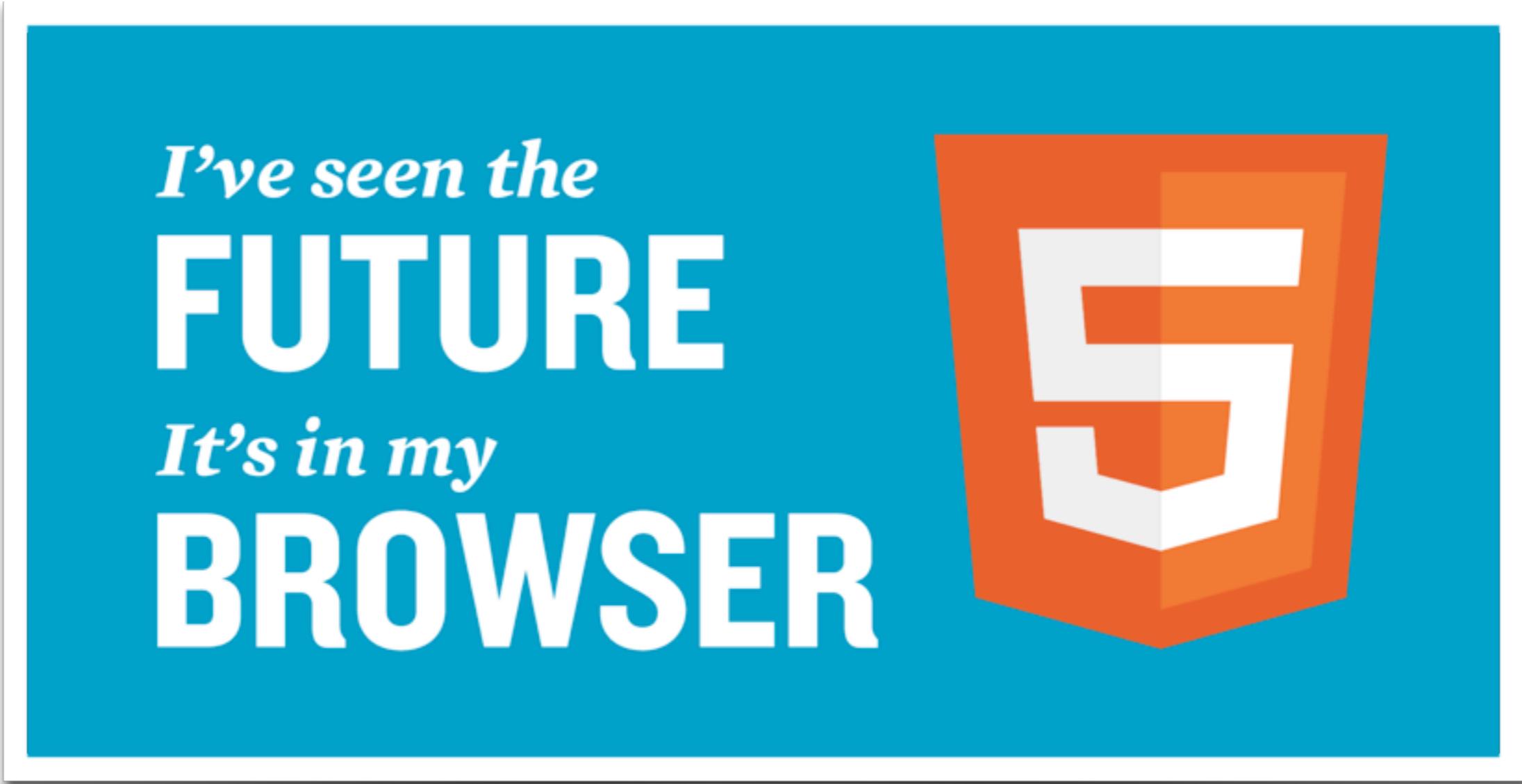
Document Your API

The screenshot shows the Swagger UI interface for the Petstore API. The URL in the browser is `petstore.swagger.wordnik.com/#!/user`. The main content area displays the `/user` endpoint with various operations listed:

- POST /user.json/createWithArray**: Creates list of users with given input array.
- POST /user.json**: Create user.
- POST /user.json/createWithList**: Creates list of users with given list input.
- PUT /user.json/{username}**: Updated user.
- DELETE /user.json/{username}**: Delete user.
- GET /user.json/{username}**: Get user by user name.
- GET /user.json/login**: Logs user into the system.
- GET /user.json/logout**: Logs out current logged in user session.

Below the `/user` section, there are links for `/pet` and `/store`, each with their own "Show/Hide", "List Operations", "Expand Operations", and "Raw" options. At the bottom, it says [BASE URL: `http://petstore.swagger.wordnik.com/api` , API VERSION: 0.2].

But if want to remain a **Web** Developer...



Modern Principles

The screenshot shows a web browser window with the following details:

- Title Bar:** Modern Principles in Web De X
- Address Bar:** blogs.atlassian.com/2012/01/modern-principles-in-web-development/
- Header:** Atlassian Home, Get Help, Plugin Exchange
- Atlassian Blogs Logo:** A stylized 'A' icon followed by the text "Atlassian Blogs".
- Search Bar:** Search Blogs with a magnifying glass icon.
- Post Title:** Modern Principles in Web Development
- Author Information:** By Rich Manalang, Developer Advocate. Includes a small profile picture of Rich Manalang, a link to his About page, and the date On January 18, 2012.
- Share Buttons:** +1 (11), Tweet (162), Like (8).
- Post Content:** A paragraph about kickstarting web apps and adjusting development principles. It mentions taking a snapshot of what's "in" today, thinking of web development phases from idea to delivery, and backing it with strong principles of building great apps.
- Principles List:** A bulleted list of core web development principles:
 - Designing for mobile first (even if you're not building a mobile app)
 - Build only single page apps
 - Create and use your own REST API
 - "Sex sells" applies to web apps

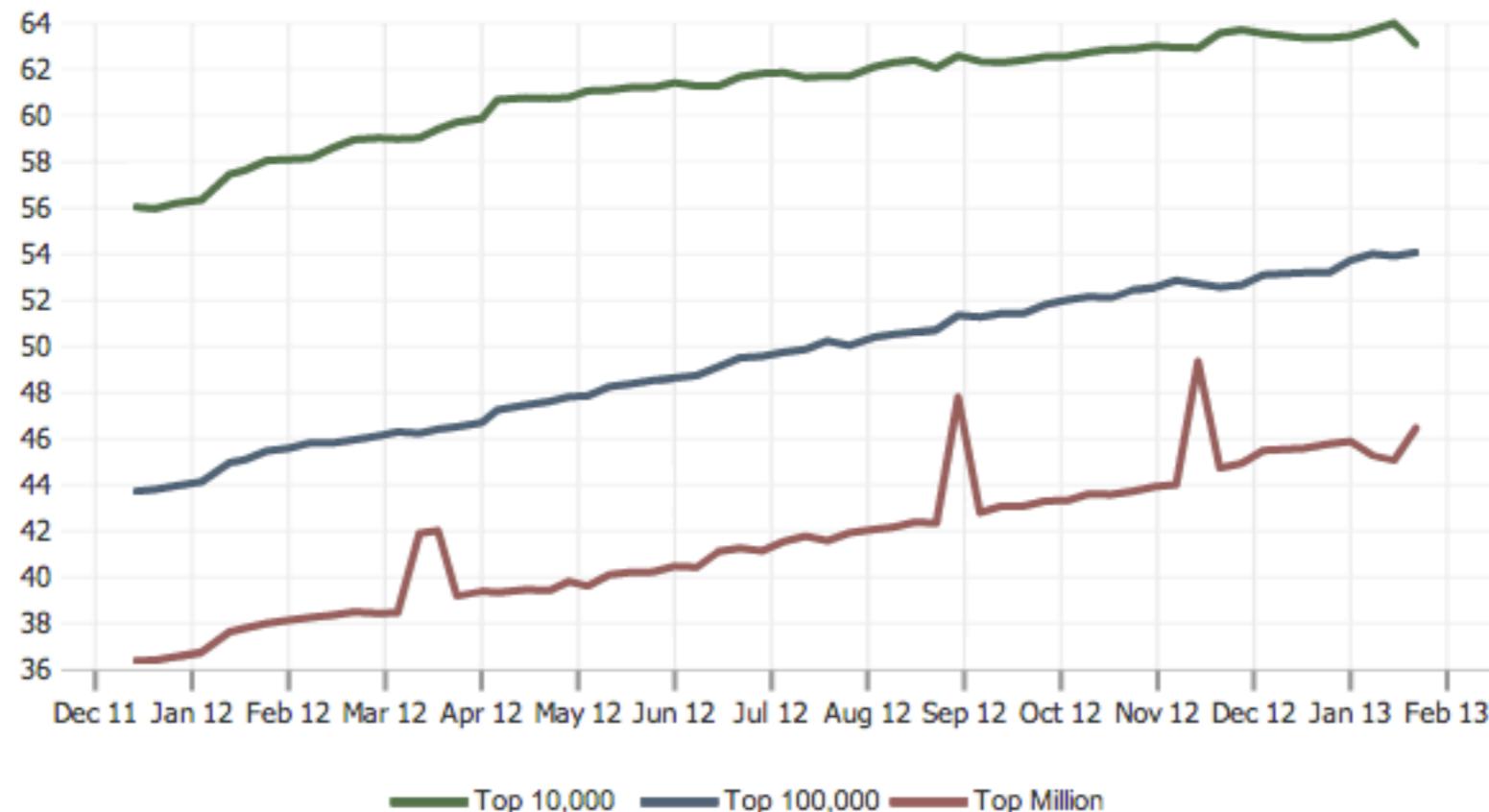
Browser Tools

- Firebug for FireFox
- Chrome Developer Tools
 - Elements & Console
 - Settings
- PageSpeed Insights
- <http://www.igvita.com/slides/2012/devtools-tips-and-tricks>



jQuery

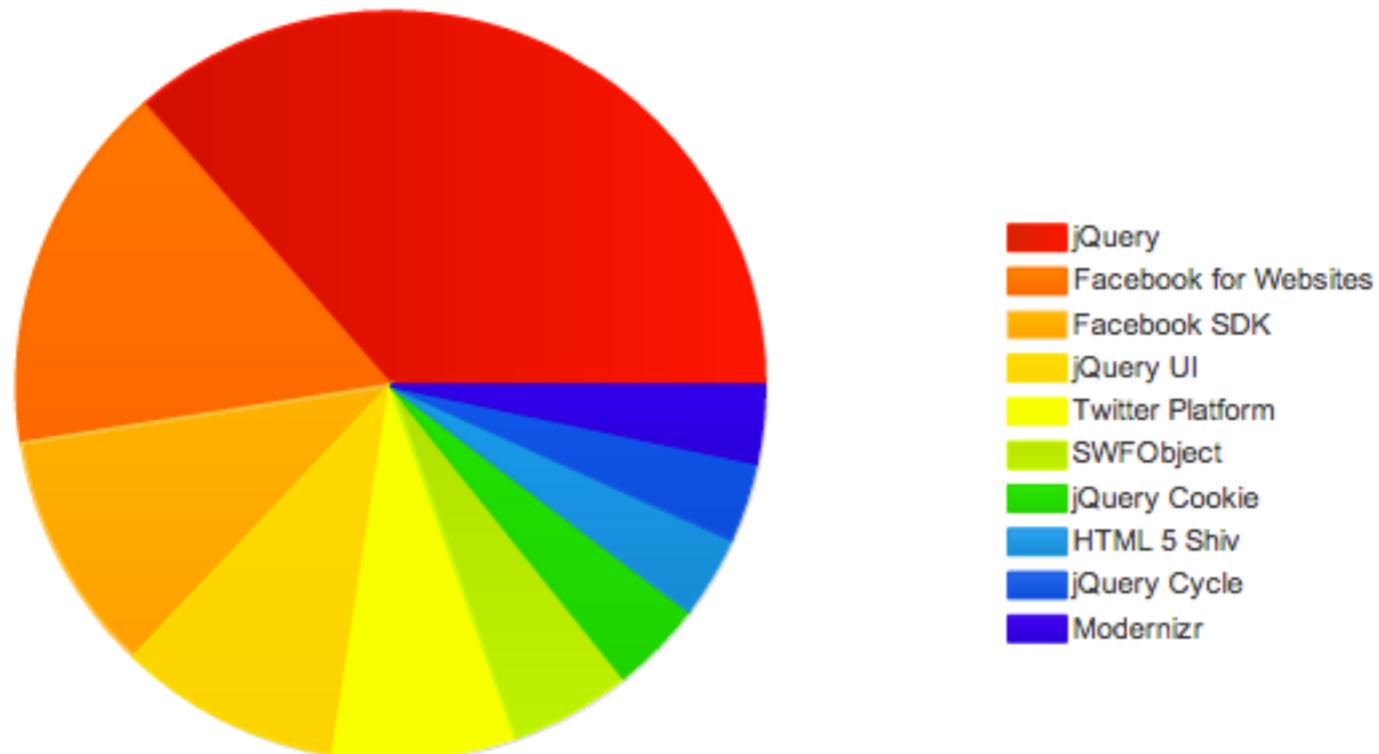
Websites using jQuery



<http://trends.builtwith.com/javascript/jQuery>

jQuery

JavaScript Distribution in Top 10,000 Sites



<http://trends.builtwith.com/javascript>

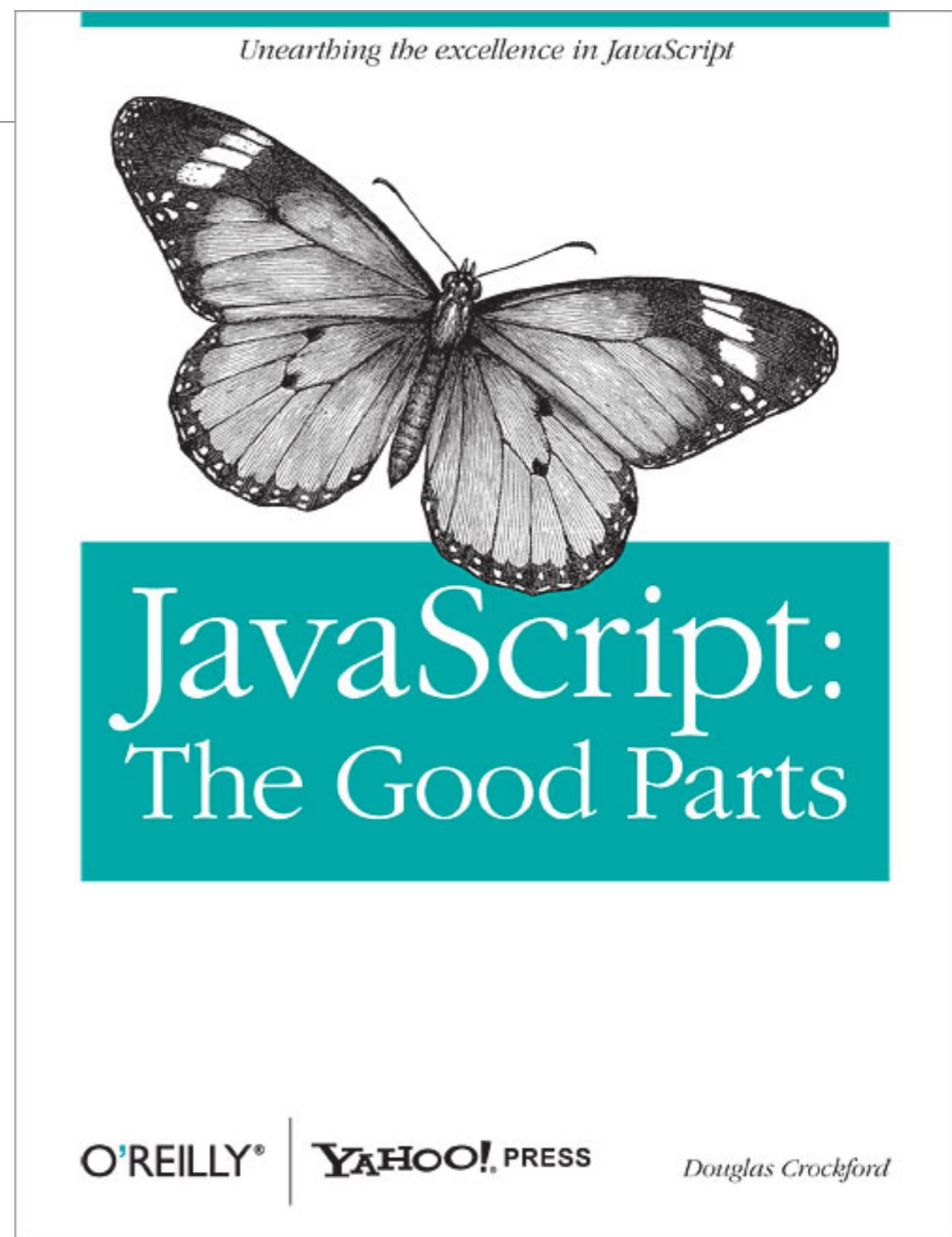
jQuery

- jQuery CDN at <http://code.jquery.com/>
- `$(document).ready`
- `$(document).on('click', '#selector', function() {})`
- jQuery UI for Dialogs, Sliders, Calendars
- `$.ajax` and `$('#div').load(url)`



JavaScript

- The Good Parts
 - Lambda
 - Dynamic Objects
 - Loose Typing
 - Object Literals



JavaScript Tips

```
"use strict";  
  
JSON.parse( text ); // converts JSON string to JavaScript Object  
JSON.stringify( obj ); // converts an Object into a serialized JSON string
```

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

JavaScript Programming Patterns

- The Old-School Way
- Singleton
- Module Pattern
- Revealing Module Pattern
- Custom Objects
- Lazy Function Definition
- <http://www.klauskomenda.com/code/javascript-programming-patterns/>

Revealing Module Pattern

```
01 var myRevealingModule = function () {  
02  
03     var privateVar = "Ben Cherry",  
04         publicVar  = "Hey there!";  
05  
06     function privateFunction() {  
07         console.log( "Name:" + privateVar );  
08     }  
09  
10    function publicSetName( strName ) {  
11        privateVar = strName;  
12    }  
13  
14    function publicGetName() {  
15        privateFunction();  
16    }  
17  
18  
19    // Reveal public pointers to  
20    // private functions and properties  
21  
22    return {  
23        setName: publicSetName,  
24        greeting: publicVar,  
25        getName: publicGetName  
26    };  
27  
28 }();  
29  
30 myRevealingModule.setName( "Paul Kinlan" );
```

CoffeeScript

JS → Coffee

Coffee → JS

/* Type here! */

```
function ($) {
  $.fn.highlight = function () {
    $(this).css({ color: 'red', background: 'yellow' });
    $(this).fadeIn();
  };
})(jQuery);
```

```
((($) ->
  $.fn.highlight = ->
    $(this).css
      color: "red"
      background: "yellow"
    $(this).fadeIn()
  ) jQuery
```



MORE INFO ↓

AngularJS

- A JavaScript MVW Framework
- From Google
- MIT Licensed
- Data-binding, Controllers, Dependency Injection
- Localization, Components, Testable
- Angular-seed for Scaffolding
- Great Documentation and Community



AngularJS Basics

The Basics

index.html

```
1. <!doctype html>
2. <html ng-app>
3.   <head>
4.     <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.0.4/
5.       angular.min.js"></script>
6.   </head>
7.   <body>
8.     <div>
9.       <label>Name:</label>
10.      <input type="text" ng-model="yourName" placeholder="Enter a name here">
11.      <hr>
12.      <h1>Hello {{yourName}}!</h1>
13.    </div>
14.  </body>
15. </html>
```

Choosing a JavaScript MVC Framework

- Journey Through The JavaScript MVC Jungle
- Learning JavaScript Design Patterns



The image shows the TodoMVC landing page. It features a large red checkmark icon followed by the text "TodoMVC" in a bold, sans-serif font. Below this, the tagline "Helping you **select** an MV* framework" is displayed. At the bottom, there are two orange buttons: "Download (1.0.1)" and "View project on GitHub". To the right of the main content area is a graphic of a clipboard with a checklist and a red pencil.

<http://addyosmani.github.com/todomvc/>

Or Just Choose AngularJS



My Feelings About AngularJS Over Time

My Angular JS Experience

3
Appointments Today
10:30 Jim Smith

4
Open Requisitions
2 On Hold

32
New Candidates
Average Age 3 Days



February 7th 2013, 11:18:08 pm

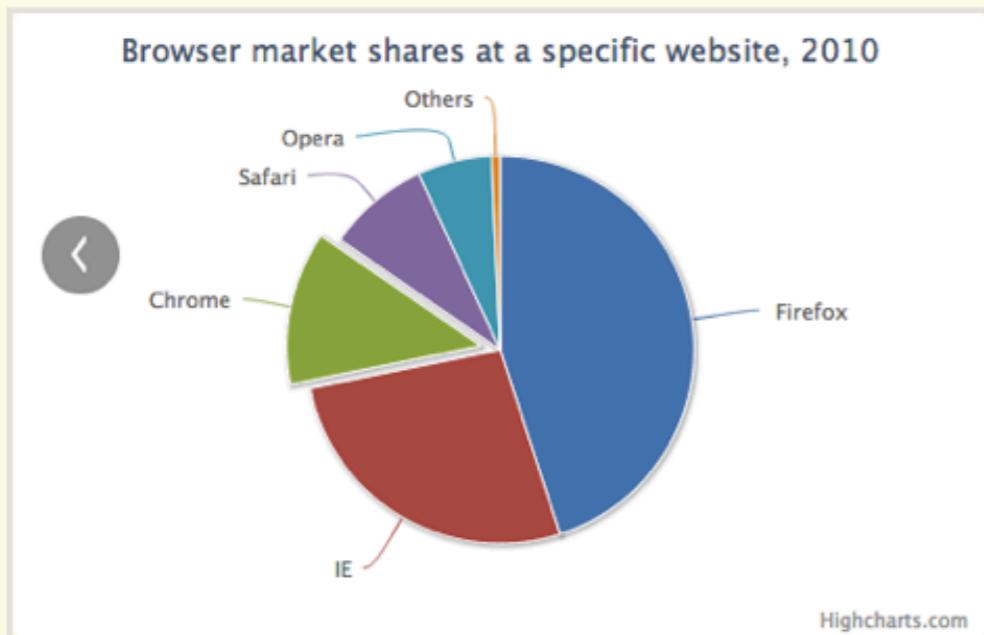
My Tasks

- Approve Requisitions **3**
- Offer Approvals **1**
- Onboard Activities **6**
- Multi Rater Request **2**

QUICK REPORTS

- Time to Fill Requisition
- Past Reviews Performance Reviews
- Overdue Reviews Performance Reviews

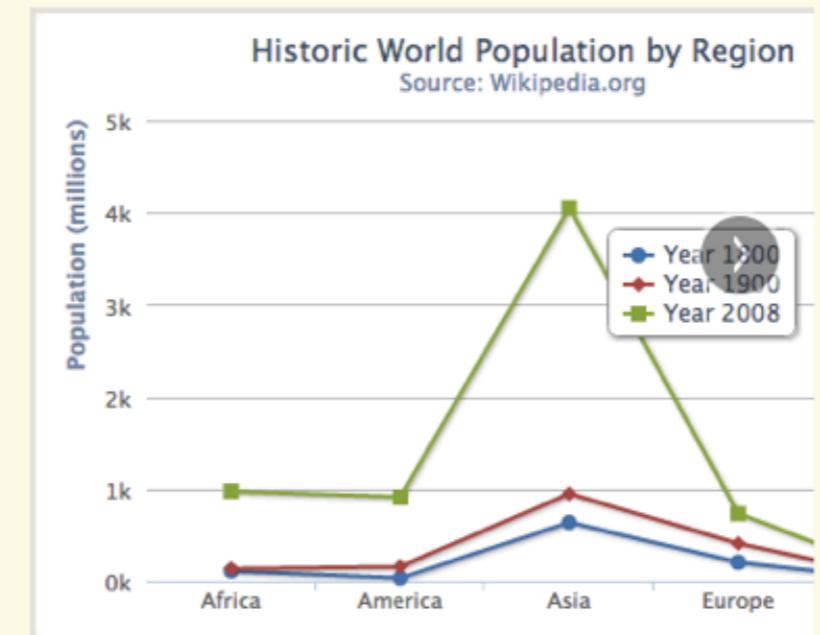
Browser market shares at a specific website, 2010



Browser	Market Share (%)
Firefox	~45%
IE	~25%
Chrome	~10%
Safari	~5%
Opera	~2%
Others	~3%

Highcharts.com

Historic World Population by Region
Source: Wikipedia.org



Region	Year 1900	Year 1900	Year 2008
Africa	~100M	~100M	~1000M
America	~100M	~100M	~1000M
Asia	~500M	~1000M	~4000M
Europe	~500M	~500M	~800M

Population (millions)

© 2013 Raible Designs

HTML5

What Does HTML5 Do?



<STORAGE>
Data can be stored on a user's computer or mobile device, so Web apps work without an Internet connection.



<TYPE>
Web pages can have flashier type with more fonts, shadows, colors and other effects.



< MOTION >
Objects move on Web pages and react to the movements of a cursor.



<GAMES>

Interactive games can run with just a Web browser without installing other software or plug-ins.



Key features of the next Web programming standard.



<VIDEO>

Video can be embedded in a Web page without a plug-in. Browser makers have not agreed on formats.



<3D>
A technology called WebGL can create interactive 3-D effects using a computer's graphics processor.



<AUDIO>
Audio is played without a plug-in. Browser makers have not agreed on formats.

How do you write HTML5?

<!DOCTYPE html>

<article> <aside> <section>
<header> <footer> <nav>
<audio> <canvas> <video>
<datalist> <details>



<applet> <center>
<frame> <frameset>



<https://developer.mozilla.org/en-US/docs/HTML/HTML5>

HTML5 Forms

```
<form>
  <input type=text placeholder=Enter your search terms>
  <input type=submit value=Search/>
</form>
```

```
<form>
  <input type=text autofocus>
  <input type=submit value=Search/>
</form>
```

```
<form>
  <input type=text required>
  <input type=submit value=Search/>
</form>
```

<http://diveintohtml5.info/forms.html>

HTML5 Forms

```
<form>
    <input type=email>
    <input type=submit value=Go>
</form>
```

HTML5 Killer Features



Editable Text



Pitbull
Don't Stop The Party

4 months

• Pop



HTML5 Killer Features



CSS3

- Animated Transitions

```
transform: rotateY(180deg);
```

- Rounded Corners

```
border-radius: 8px 8px 0 0;
```

- Drop Shadows

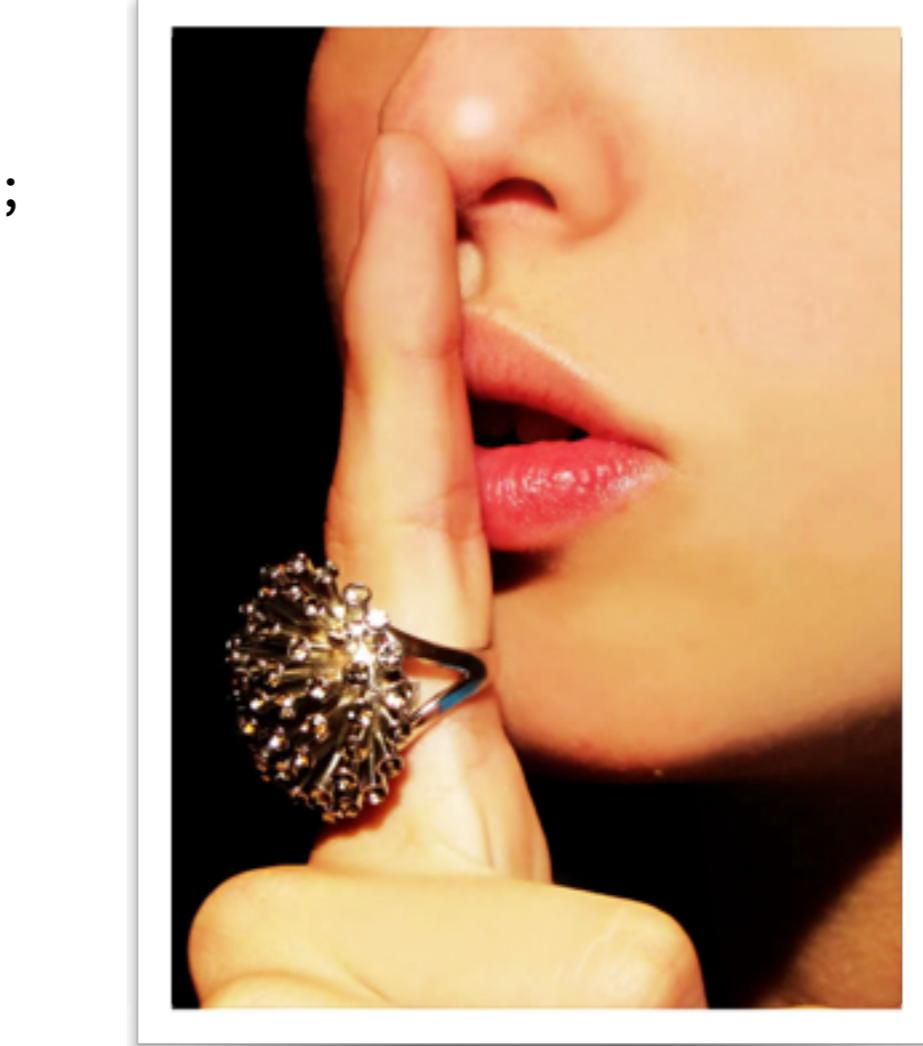
```
box-shadow: 2px 2px 4px 4px;
```

- Gradient Colors

- Styling based on sibling count

- More cursors for better usability

- Custom Checkboxes and Radio Buttons



<http://lea.verou.me/css3-secrets>

CSS3 Media Queries

```
/* Smartphones (portrait and landscape) ----- */
@media only screen and (min-device-width: 320px) and (max-device-width: 854px) {
    body {
        padding: 10px;
    }

    .pubbox textarea {
        width: 90%;
    }

    .width160 {
        width: 100px;
        min-width: 100px;
    }
}

/* iPad (portrait and landscape) ----- */
@media only screen and (min-device-width: 768px) and (max-device-width: 1024px) {
    body {
        padding-top: 50px;
    }
}
```

Cool HTML5 Demos

<http://fff.cmiscm.com>



Bootstrap

- Good-looking websites by default
- Layouts, navs, pagination, buttons
- Responsive for Mobile
- 13 Custom jQuery Plugins
- Themes
 - {wrap}bootstrap (Market)
 - Bootswatch (OSS)



LESS



```
// LESS                                /* Compiled CSS */

@color: #4D926F;

#header {
  color: @color;
}
h2 {
  color: @color;
}
```

LESS

```
// LESS                                     /* Compiled CSS */

.rounded-corners (@radius: 5px) {
  border-radius: @radius;
  -webkit-border-radius: @radius;
  -moz-border-radius: @radius;
}

#header {
  .rounded-corners;
}

#footer {
  .rounded-corners(10px);
}
```

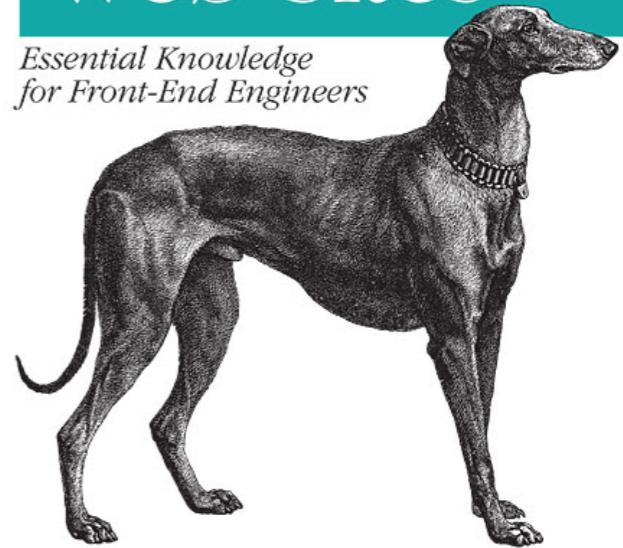
High Performance Web Sites

1. Make Fewer HTTP Requests
2. Use a Content Delivery Network
3. Add Expires Headers
4. Gzip Components
5. Put Stylesheets at the Top
6. Put Scripts at the Bottom
7. Avoid CSS Expressions

14 Steps to Faster-Loading Web Sites

High Performance Web Sites

*Essential Knowledge
for Front-End Engineers*

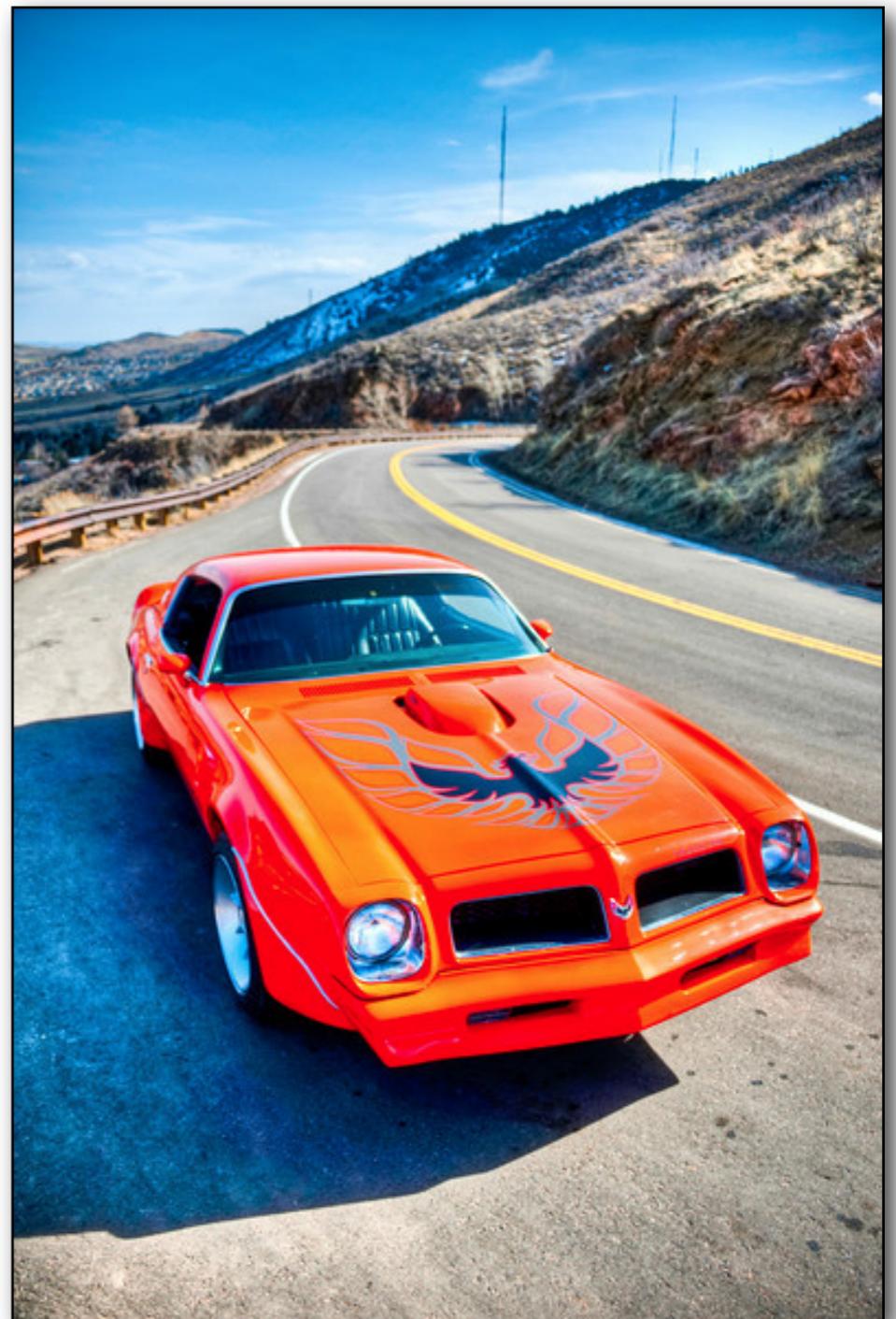


O'REILLY®

*Steve Souders
Foreword by Nate Koechley*

High Performance Web Sites

8. Make JavaScript and CSS External
9. Reduce DNS Lookups
10. Minify JavaScript
11. Avoid Redirects
12. Remove Duplicates Scripts
13. Configure ETags
14. Make Ajax Cacheable



My Page Speed Experience

- Biggest Problem is HTTP Requests
 - Minify and Concatenate
- GZipping has biggest score gain
- Expires Headers for Browser Caching
 - Use Apache or Nginx
- Image Sprites for CSS
 - <http://spritecow.com> is great



Wro4j

- Open Source Java project for optimization of web resources
- Provides concatenation and minimization of JS and CSS
- Gzip, YUI Compressor, JsHint, JsHint, CssLint, LESS, SASS, CoffeeScript, Dojo Shinksafe



Sidenote: WebJars and UrlRewrite Filter

- WebJars: client-side web libraries packages in JARs
 - <http://webjars.org>
 - Servlet 3, Play 2, Grails, Dropwizard, Spring MVC, Tapestry and Wicket
- UrlRewriteFilter: like Apache's mod_rewrite
 - <http://tuckey.org/urlrewrite>
 - Clean URLs and just about everything else
- Spring MVC's <default-servlet-handler/> [is your friend](#)

Techniques

- Versioning Static Assets with UrlRewriteFilter
 - [http://raibledesigns.com/rd/entry/
versioning static assets with urlrewritefilter](http://raibledesigns.com/rd/entry/versioning_static_assets_with_urlrewritefilter)
- Adding web resource fingerprinting to AppFuse with wro4j
 - <http://www.operatornew.com/2012/10/adding-web-resource-fingerprinting-to.html>
- Improving AppFuse's PageSpeed with Apache
 - [http://raibledesigns.com/rd/entry/improving appfuse s pagespeed with](http://raibledesigns.com/rd/entry/improving_appfuse_s_pagespeed_with)

But what about nginx?

- An open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server
- Powers Netflix, Wordpress.com, GitHub and Heroku
- <http://kevinwashington.com/nginx-for-mac-os-x-mountain-lion-in-2-minutes/>



Apache Gzip and Expires Headers

- mod_pagespeed - <https://developers.google.com/speed/pagespeed/mod>
 - Automatically applies web performance best practices w/o modification
- Improving AppFuse's PageSpeed with Apache
 - Configured mod_deflate, mod_expires and turned on KeepAlive
 - PageSpeed went from 24 to 96!
 - YSlow went from 90 to 98



/etc/httpd/conf.d/deflate.conf

```
<IfModule mod_deflate.c>

    SetOutputFilter DEFLATE

        AddOutputFilterByType DEFLATE text/plain text/html text/xml
text/css application/xml application/xhtml+xml application/rss
+xml application/javascript application/x-javascript

        DeflateCompressionLevel 9

        BrowserMatch ^Mozilla/4 gzip-only-text/html
        BrowserMatch ^Mozilla/4\.\.0[678] no-gzip
        BrowserMatch \bMSIE !no-gzip !gzip-only-text/html

        DeflateFilterNote Input instream
        DeflateFilterNote Output outstream
        DeflateFilterNote Ratio ratio
        LogFormat '%r %{outstream}n/%{instream}n (%{ratio}n%%)'
deflate
</IfModule>
```

/etc/httpd/conf.d/expire.conf

```
<IfModule mod_expires.c>
    ExpiresActive On
    <FilesMatch \.(jpe?g|png|gif|js|css)$>
        ExpiresDefault access plus 1 week
    </FilesMatch>
    ExpiresByType image/jpeg access plus 1 week
    ExpiresByType image/png access plus 1 week
    ExpiresByType image/gif access plus 1 week
    ExpiresByType text/css access plus 1 week
    ExpiresByType application/javascript access plus 1 week
    ExpiresByType application/x-javascript access plus 1 week
</IfModule>
```

Application Architecture

- Server to Client
- Performance implications
 - Twitter
 - Airbnb
 - Charm
- Ajaxified Body with pjax

<https://github.com/defunkt/jquery-pjax>



Mobile Devices

- If developing a mobile app with web technologies
 - Use PhoneGap or Sencha Touch
- Otherwise, add a **viewport** meta tag



```
<meta name=viewport content=width=device-width,  
initial-scale=1.0>
```

Mobile Devices - CSS3 Media Queries



```
/* Smartphones (portrait and landscape) -----  
*/  
@media only screen and (min-device-width: 320px)  
and (max-device-width: 854px) {  
    /* Smartphone rules */  
}  
  
/* iPad (portrait and landscape) ----- */  
@media only screen and (min-device-width: 768px)  
and (max-device-width: 1024px) {  
    /* Tablet rules */  
}
```

Mobile Devices - Hide Address Bar

```
<script type="text/javascript">
    // Hide address bar for smartphones
/Mobile/.test(navigator.userAgent) && !location.hash
    && setTimeout(function () {
        if (!pageYOffset) window.scrollTo(0, 1);
    }, 1000);
</script>
```

Mobile Devices - Disable Focus Zoom

```
(function(doc) {
  var addEvent = 'addEventListener',
      type = 'gesturestart',
      qsa = 'querySelectorAll',
      scales = [1, 1],
      meta = qsa in doc ? doc[qsa]('meta[name=viewport]') : [];

  function fix() {
    meta.content = 'width=device-width,minimum-scale=' +
      scales[0] + ',maximum-scale=' + scales[1];
    doc.removeEventListener(type, fix, true);
  }

  if ((meta = meta[meta.length - 1]) && addEvent in doc) {
    fix();
    scales = [.25, 1.6];
    doc[addEvent](type, fix, true);
  }
}(document));
```

The Cloud



heroku

Heroku for Java
by Adam - Aug 25, 2011

We're pleased to announce the public beta of Heroku for Java. Java is the fourth official language available on the [Cedar](#) stack.

Java is, by [many measures](#), the world's most popular programming language. In addition to its large and diverse developer base, it offers a huge ecosystem of libraries and tools, an extremely well-tuned VM for fast and reliable runtime performance, and an accessible C-like syntax.

But there are also many criticisms commonly leveled against the language. We'll take a closer look at Java's strengths and weaknesses in a moment, but first:

Heroku for Java in 2 minutes



The Cloud



- Supports Ruby, Node.js, Clojure, Java, Python and Scala
- Java: <https://devcenter.heroku.com/articles/java>

```
$ git push heroku master
Counting objects: 47, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (25/25), done.
Writing objects: 100% (47/47), 10.25 KiB, done.
Total 47 (delta 19), reused 42 (delta 17)

-----> Heroku receiving push
-----> Java app detected
-----> Installing OpenJDK 1.6... done
-----> Installing Maven 3.0.3... done
-----> Installing settings.xml... done
-----> executing /app/tmp/repo.git/.cache/.maven/bin/mvn -B -Duser.home=/tmp/build_3k0p14ghrmzs
[INFO] Scanning for projects...
```

The Cloud



- Supports Spring, Grails, Scala, Play, Node.js, Ruby/Rails/ Sinatra
- Services: MySQL, PostgreSQL, MongoDB, Redis, RabbitMQ
- Biggest con: no pricing yet

The Cloud



Powered by the industry's
first Cloud Application Hypervisor

Ravello enables you to
Develop in the cloud.
Deploy anywhere

Apache TomEE



Security Matters

- Be aware of SQL and Content Injection
- Disable Cross-Site Scripting (XSS)
- Don't give too much information in error pages and exceptions
- Stop unvalidated redirects and forwards
- Always use https if your application has private data
- Understand what browsers have to deal with



The Modern **JVM** Web Developer

- Starts with Fast Hardware
- Uses IntelliJ IDEA
- Leverages jQuery, HTML5, and CSS3
- Creates High Performance Web Sites
- For Mobile Devices
- In the Cloud
- And *cares* about Security



Staying Modern

- Read
- Attend Conferences
 - Submit a talk!
- Write
- Do
 - Get Paid 
 - Open Source

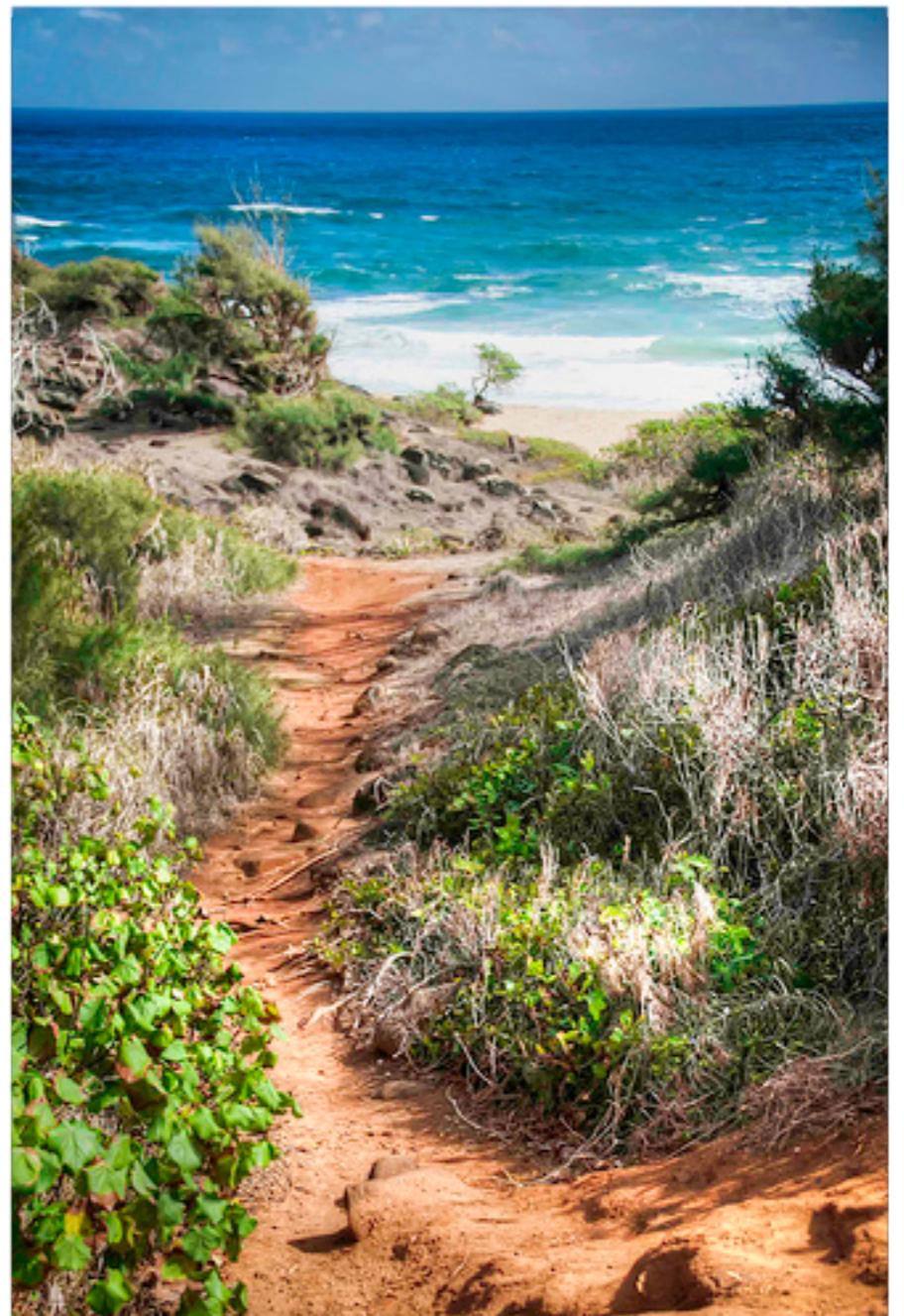


Learning Resources

- <http://learn.shayhowe.com>



- <http://teamtreehouse.com>



Cask-Strength HTML5 Training

- The Modern Web Developer's Toolkit - Scott Davis
- Advanced JavaScript: The 'What You Need To Know' Parts - Kyle Simpson
- Airplane-Mode HTML5: Mobile Web Development for Tablets and Smartphones - Scott Davis
- Unleash The Faster Web: Secrets of Website Performance - Kyle Simpson
- Writing HTML5 Games: 2D Game Programming Techniques - Dave Geary
- Register at <http://thirstyhead.eventbrite.com>; Denver in March, Austin in April



Questions?

Contact Information

<http://raibledesigns.com>

[@mraible](#)

Presentations



<http://slideshare.net/mraible>

Code



<http://github.com/mraible>

