# Pre-Spec Work Part 2 - Know Thy Medium

CSE 112

Winter 2016

Prof. Powell

## Tools and Tech First?

- Sometimes we do start with tools or tech first
- Corporate Incumbency / Technical Legacy
  - “We’re a Java shop”
- Hopefully tool/tech is driven by appropriate need but if we are honest they aren’t far too often
  - “Everybody uses jQuery”, “Rails is hot!”, “We must go to the cloud”, “Let’s do an App”, “Angular sucks, react rocks”, etc.

## Loving Constraints

- Constraints are good
  - They make our job easier because they are fixed points
  - Open ended pursuit of perfect tech/tool answers can lead to analysis paralysis
- Reviewing our constraints
  - Web based, JavaScript, MEAN stack, try to be modern SE using proper tools

## Web Based

- Browser based as opposed to app based
- Open distribution but discovery challenges
- Avoid install issues, but less persistence
- Speed/Quality concerns
  - Discuss
- Q: Is this choice appropriate if we consider our project type?

## My Environmental Assumptions

*"To succeed in the Web environment you must assume the Internet is a hostile, unpredictable network, populated by users with widely varying skill levels and intentions."*

## There are no guarantees

- You are not guranteed:
  - High Bandwidth
  - Low latency
  - Predictable error free delivery
  - Safe data (data may be not just surprising but malicious)
  - A client-side with appropriate version/tech
  - A client-side configured or running properly
  - Friendly end-users (do bad things not just by mistake)
  - Technically aware end-users

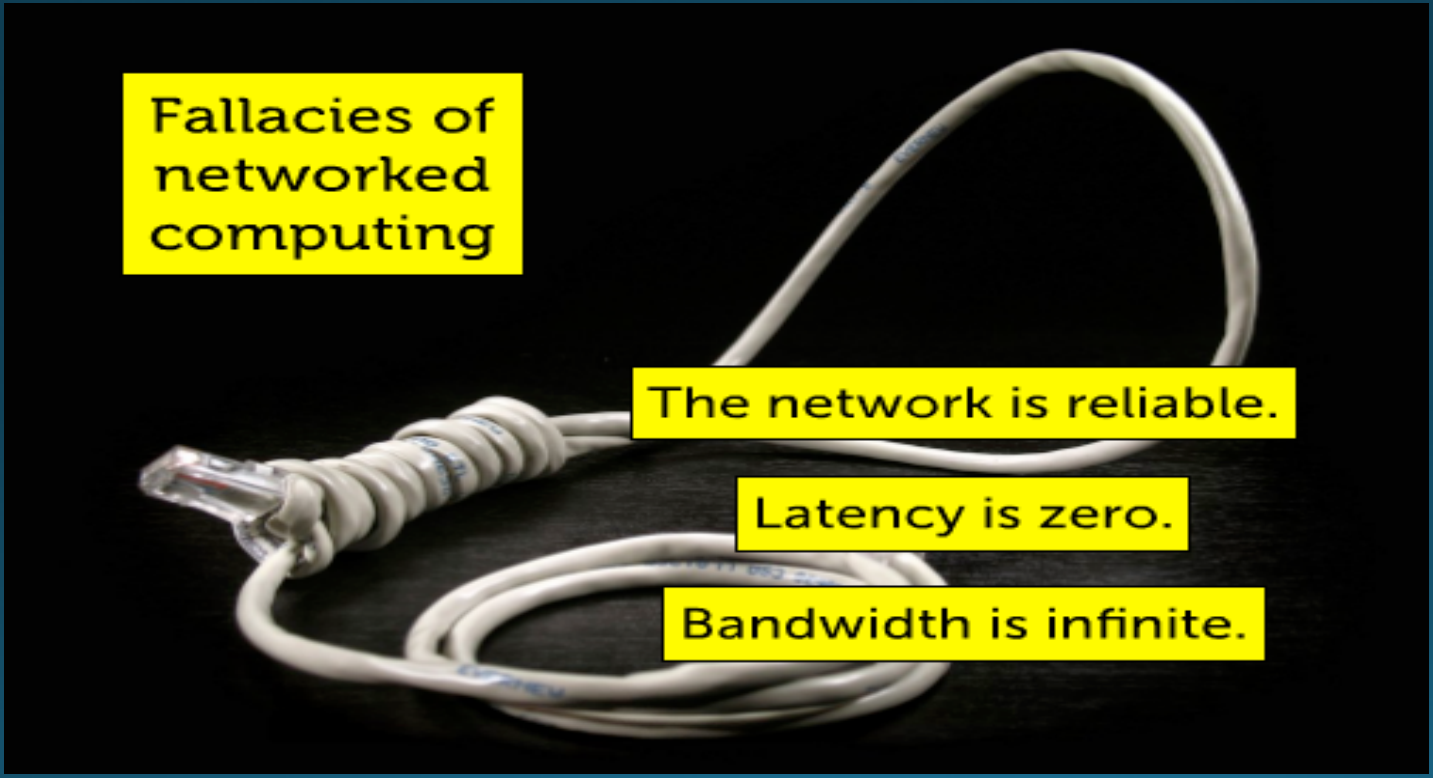
## It’s not just me

Douglas Crockford on browsers:

**“The most hostile software development environment imaginable.”**

*Source: http://nate.koechley.com/talks/2008/at-media-2008-pro-frontend-engineering.pdf*

## Network computing is hard

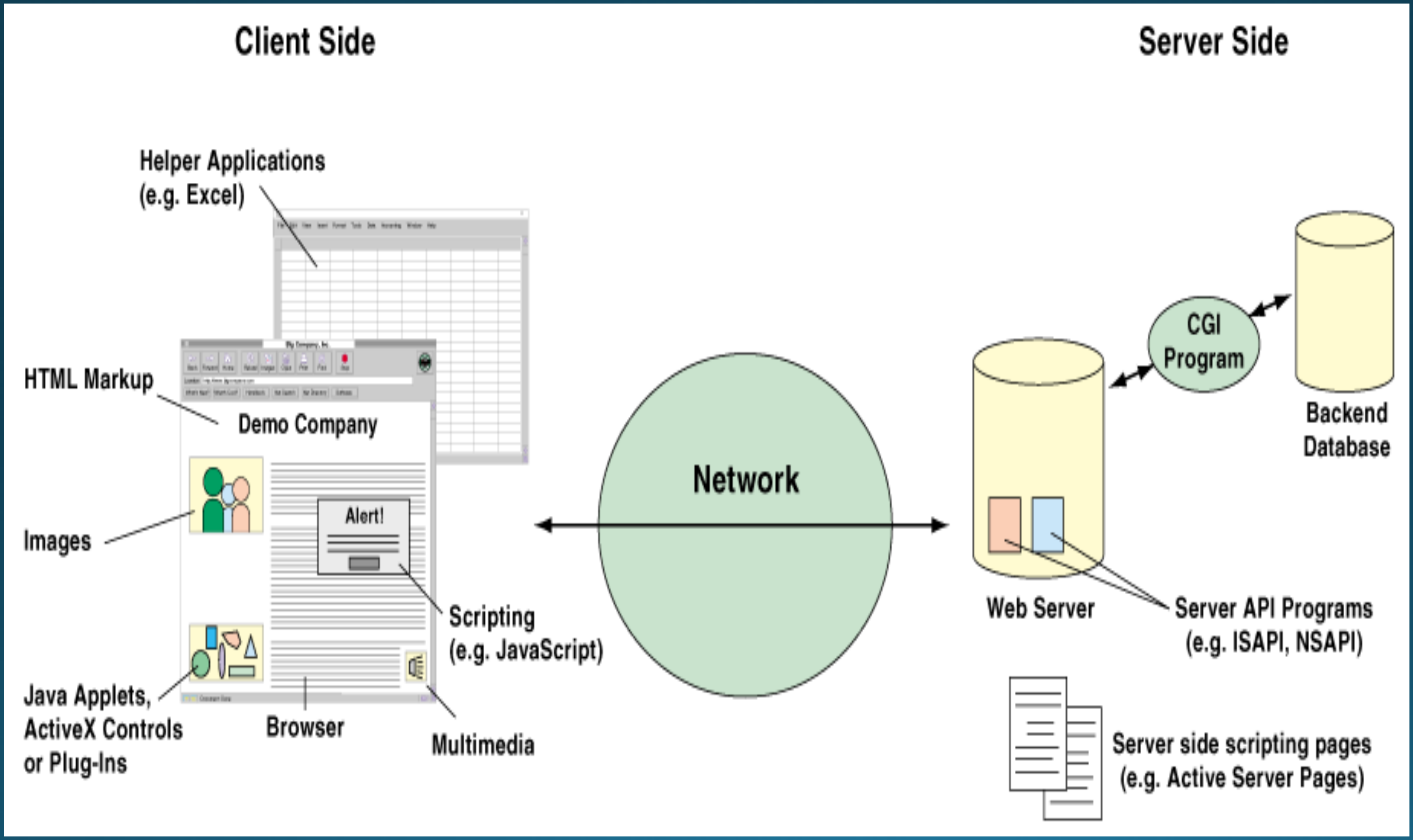


## Environmental Counterpoint

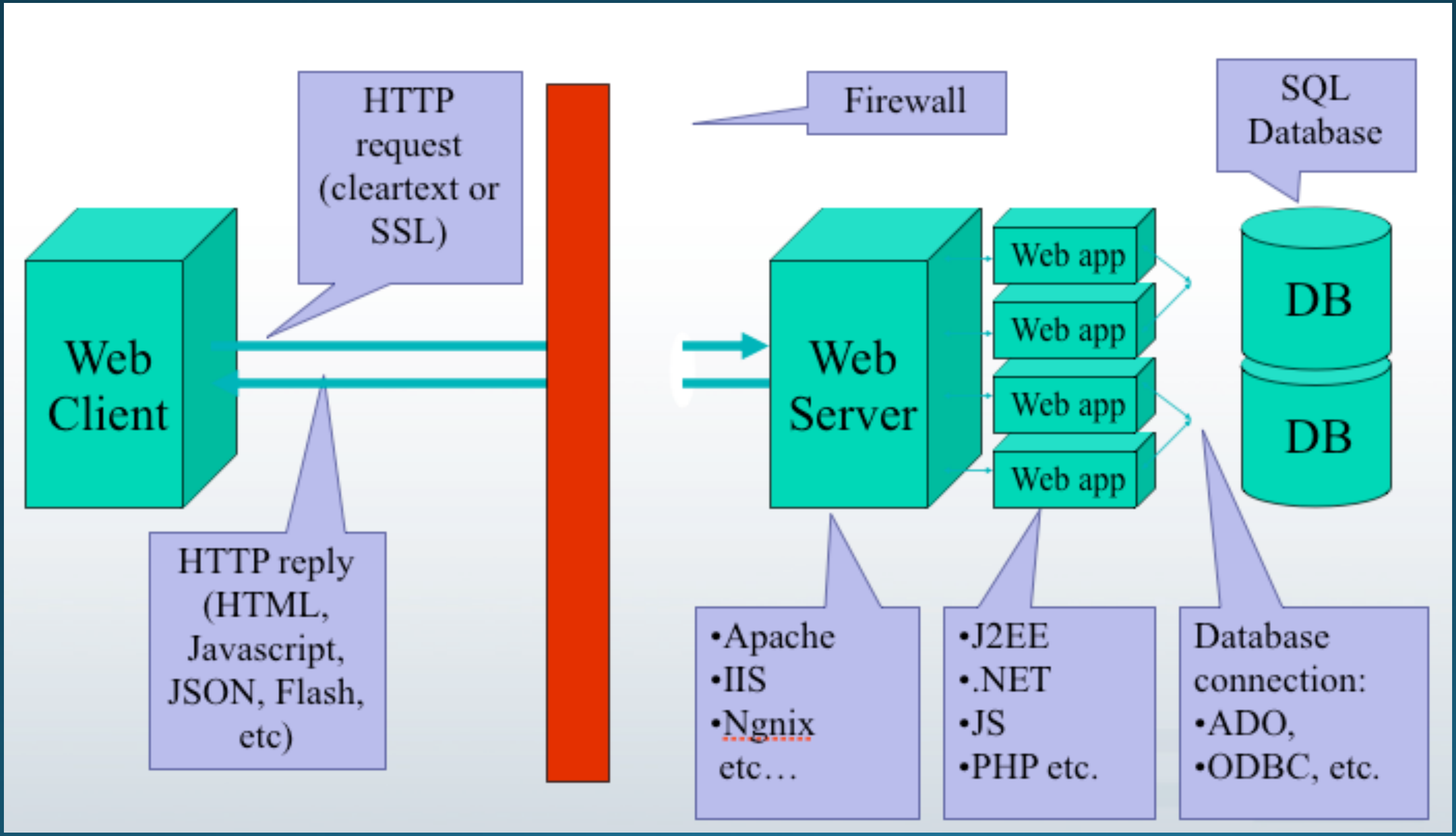
Am I wrong and just being too cynical?
Observation says probably not:

- “Layer 8” Error Correction
- Unstable Browser as platform
- Unstable tech stack and standards
- Actual network monitoring esp. RUM JS based
- Observed hacking
- Our video of users

# Development Medium


Client Side
Server Side
Helper Applications (e.g. Excel)
HTML Markup
Demo Company
Alert!
Images
Java Applets, ActiveX Controls or Plug-Ins
Browser
Scripting (e.g. JavaScript)
Multimedia
Network
Web Server
CGI Program
Backend Database
Server API Programs (e.g. ISAPI, NSAPI)
Server side scripting pages (e.g. Active Server Pages)

## Web App in Context of Medium



## Client-Side/Server-Side Trade-offs

- Client-side is fast, but it is unpredictable, insecure and unsafe
- Server-side is predictable but it is slow because of network round-trip
- Clear idea is to balance both sides rather than suggest a sole side solution

## Thin and Fat Clients

- Thin-client: very safe, but not-rich and slow
- Fat-client: very rich and fast, but likely fragile and insecure
- If we could we would engineer a fat client solution to error gracefully or fall back to thin client version in less than ideal situations (tech, network, hack)

## Web Toolbox

| **Client Side** | **Server Side** |
|---|---|
| Helper Applications | CGI scripts and programs |
| Browser API Programs<br>* Netscape Plugins<br>* ActiveX Controls<br>* Google's Native Client | Server API Programs<br>* ISAPI * NSAPI * Apache Modules |
| Java Applets | Java Servlets |
| Scripting Languages<br>* JavaScript<br>* ~~VBScript~~ | Scripting / Programming Frameworks<br>* JSP * ASP.NET * Classic Active Server Pages (ASP) * ColdFusion * PHP<br>* Ruby (Rails) * JavaScript Note: Nodejs somewhat out of box really |

## General Tool Trade-off

- Complexity and Speed
- Abstraction and Ease

- Higher/Lower Up ~ Speed
- Difficulty of coding ~ Expense
- Higher/Lower Up ~ Control

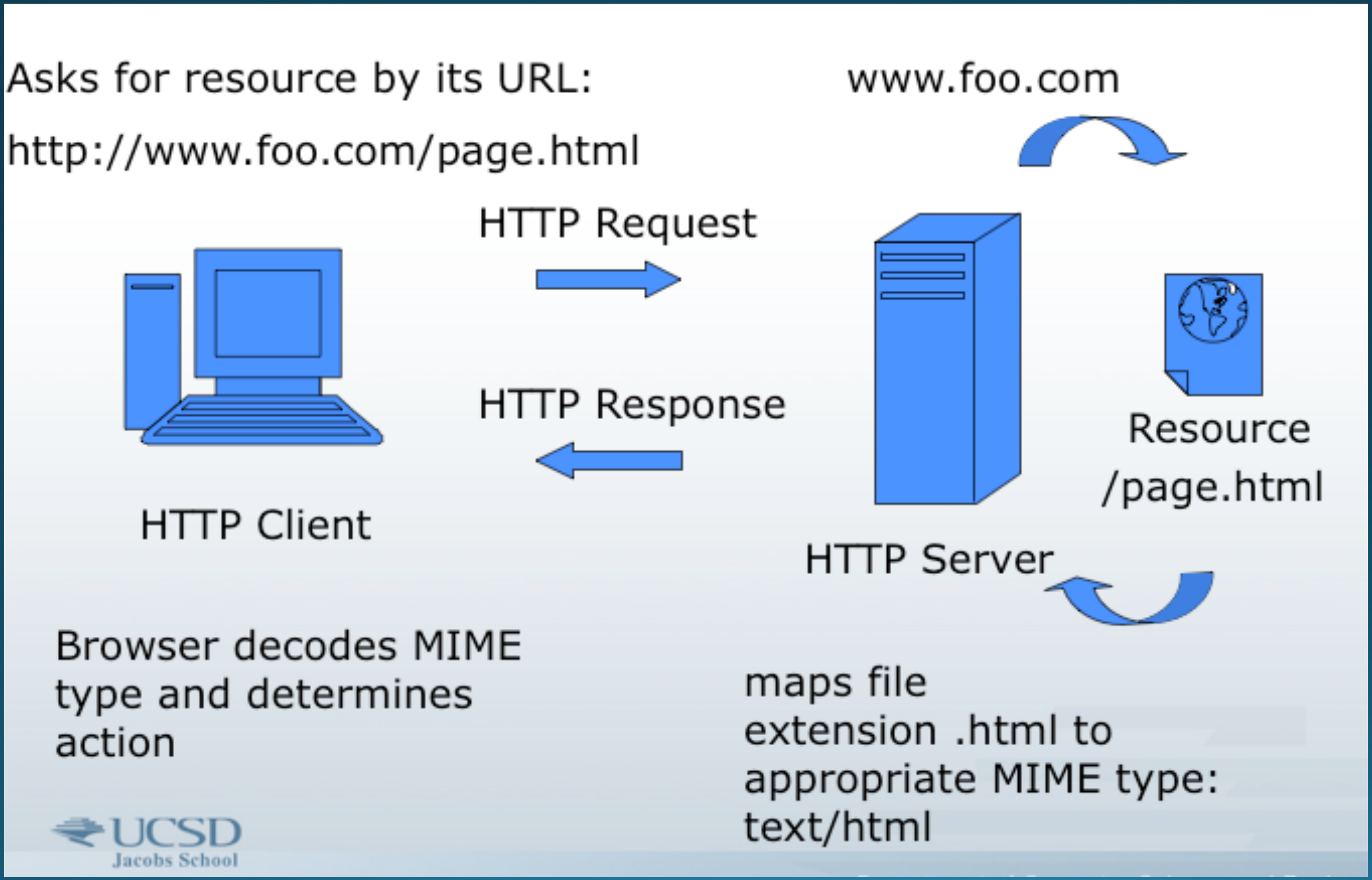
## Tool Sanity

- Rough functional & trade-off equivalences within toolbox bins, but executional, ecosystem, and other differences
- General performance, feature or quality of two items in a toolbox bin is executional rather than architectural
- Scale is generally not language issue, that is app architecture
- One can be Web scale (or Web stupid) in anything

## HTTP is the Key

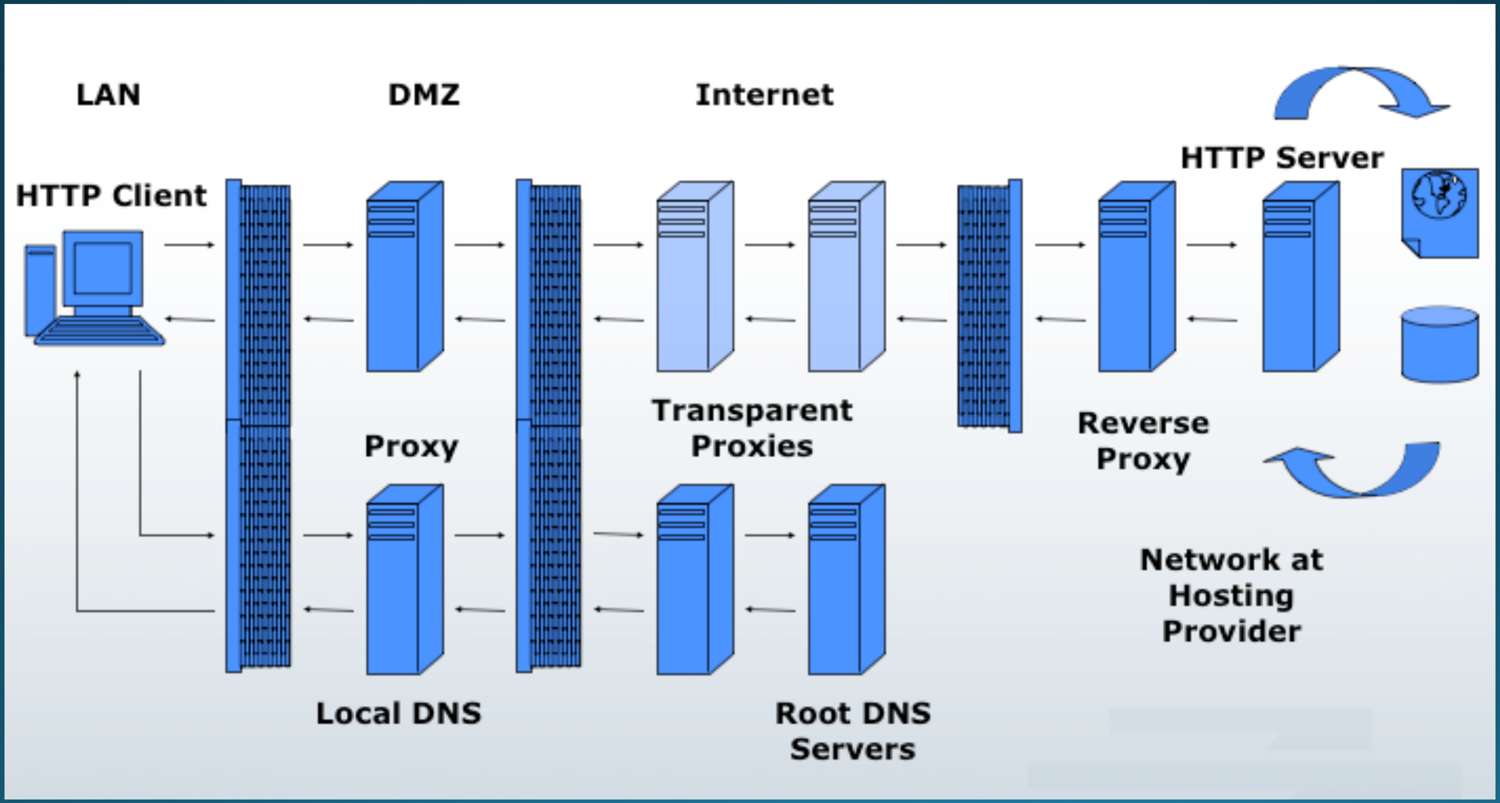
- Sure markup (HTML), look (CSS), coding (JS), browsers, etc. are unique but the part that really makes this vastly different from programming (as is in desktop) is the network
- Theory: Knowing HTTP is key to an informed understanding of robust Web programming
- HTTP - simple stateless application layer protocol

## Basic HTTP Request/Response Cycle

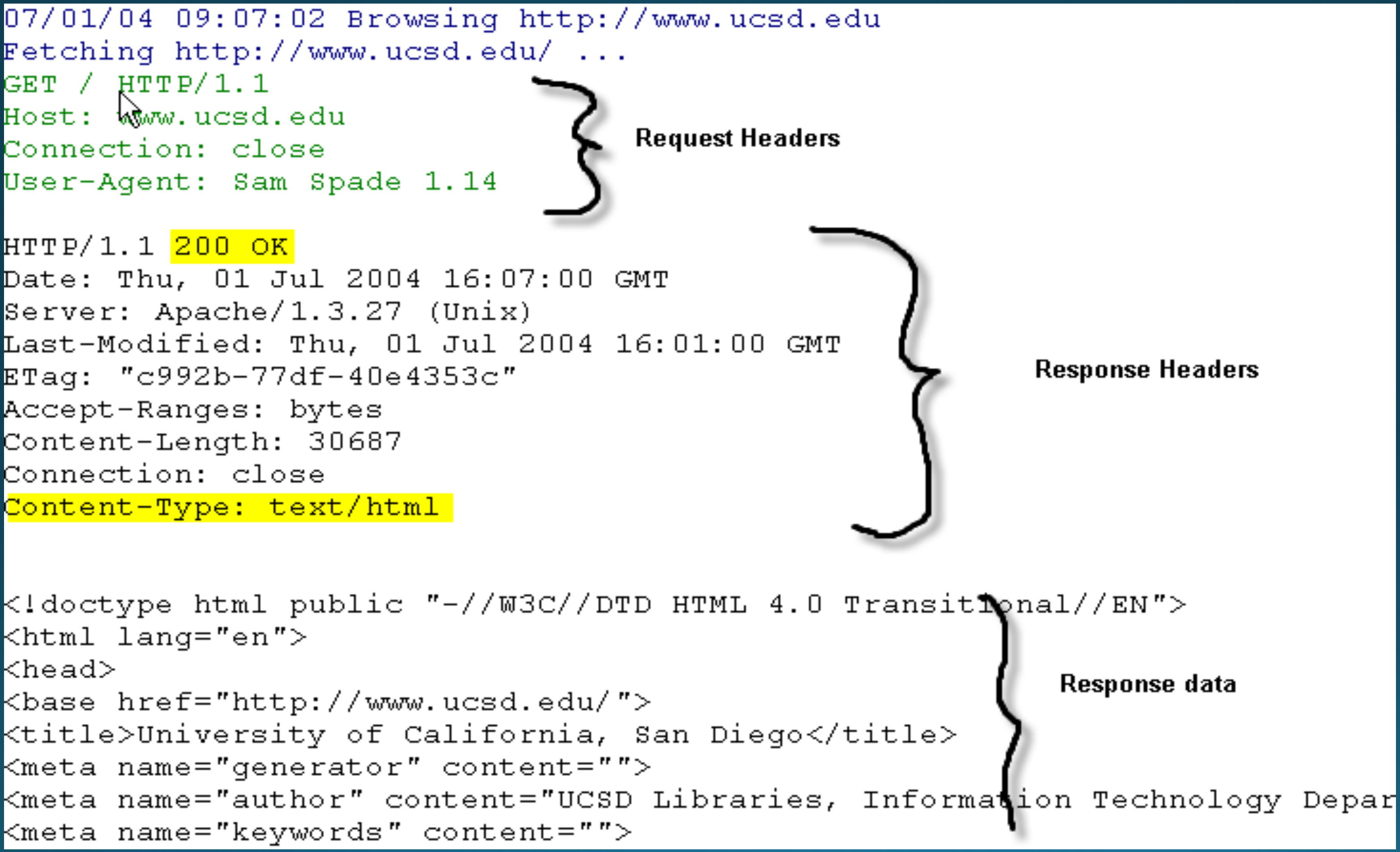

Asks for resource by its URL:
http://www.foo.com/page.html
www.foo.com
HTTP Request
HTTP Response
HTTP Client
HTTP Server
Resource
/page.html
Browser decodes MIME type and determines action
maps file extension .html to appropriate MIME type: text/html
UCSD
Jacobs School

## That was too simple

Reality actually is more like...



# HTTP Up Close

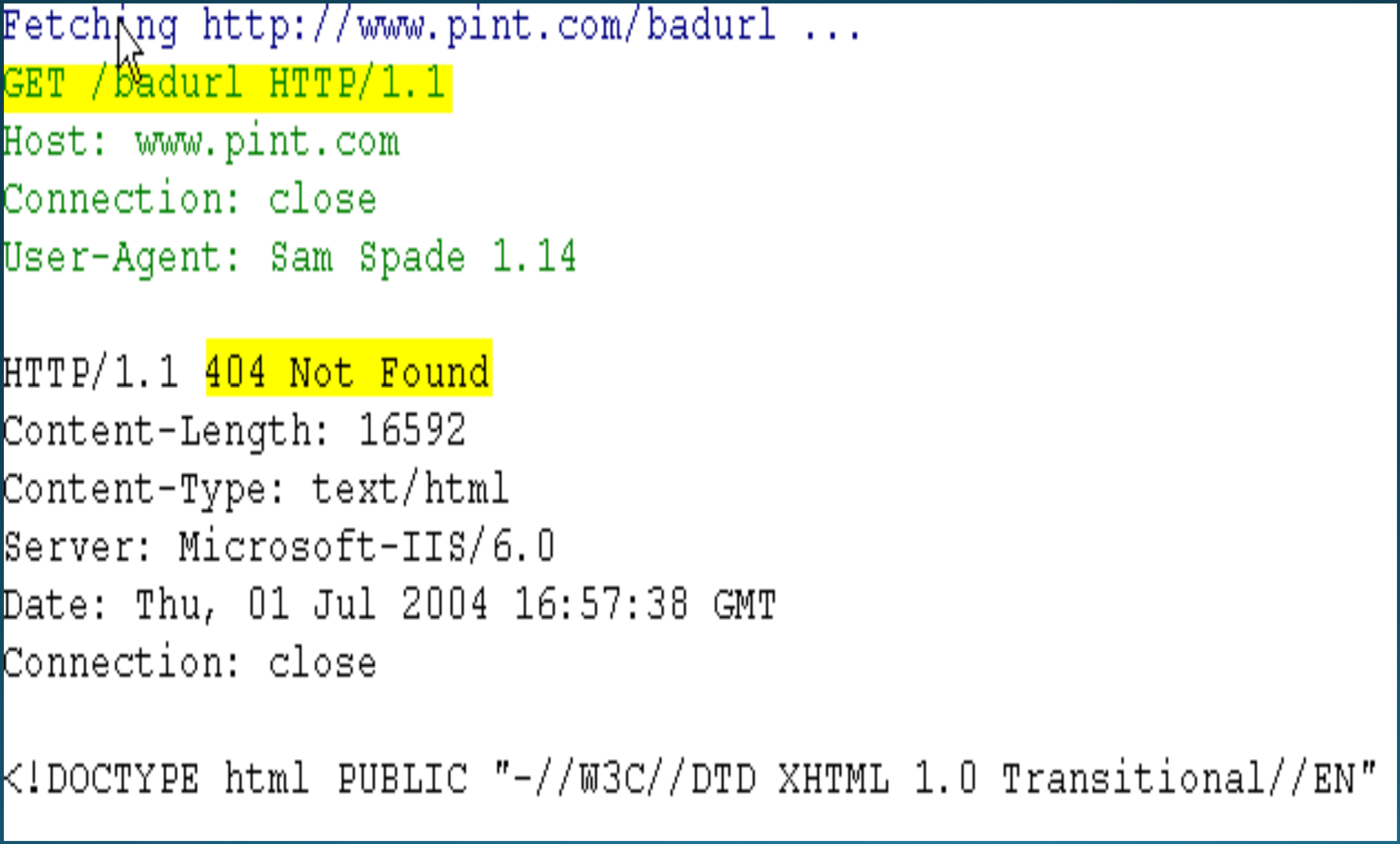


```
07/01/04 09:07:02 Browsing http://www.ucsd.edu
Fetching http://www.ucsd.edu/ ...
GET / HTTP/1.1
Host: www.ucsd.edu
Connection: close
User-Agent: Sam Spade 1.14

HTTP/1.1 200 OK
Date: Thu, 01 Jul 2004 16:07:00 GMT
Server: Apache/1.3.27 (Unix)
Last-Modified: Thu, 01 Jul 2004 16:01:00 GMT
ETag: "c992b-77df-40e4353c"
Accept-Ranges: bytes
Content-Length: 30687
Connection: close
Content-Type: text/html


<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="en">
<head>
<base href="http://www.ucsd.edu/">
<title>University of California, San Diego</title>
<meta name="generator" content="">
<meta name="author" content="UCSD Libraries, Information Technology Depar
<meta name="keywords" content="">
```

## HTTP Up Close #2



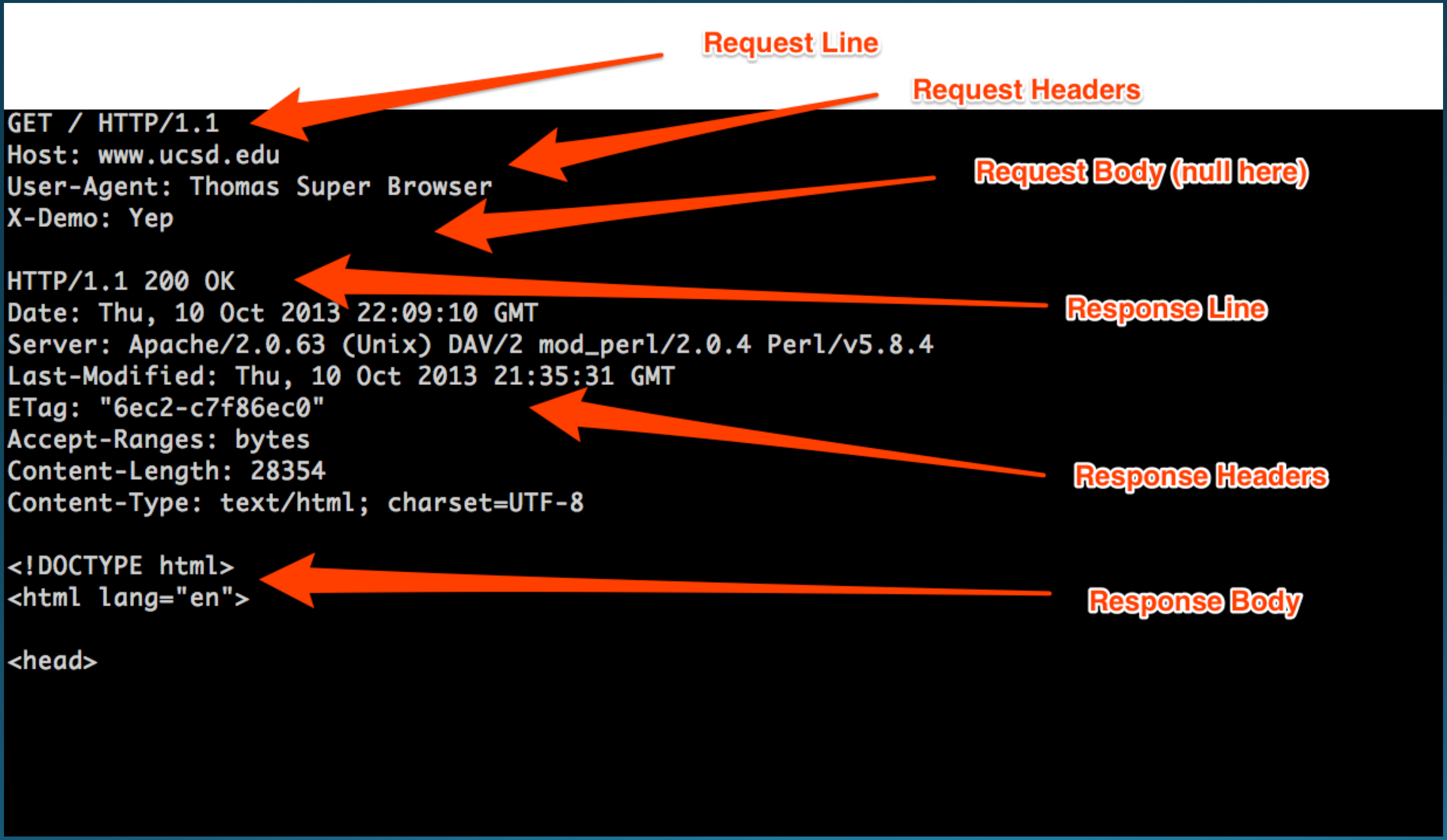
```
Fetching http://www.pint.com/badurl ...
GET /badurl HTTP/1.1
Host: www.pint.com
Connection: close
User-Agent: Sam Spade 1.14

HTTP/1.1 404 Not Found
Content-Length: 16592
Content-Type: text/html
Server: Microsoft-IIS/6.0
Date: Thu, 01 Jul 2004 16:57:38 GMT
Connection: close

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

## Law of 3

Focus on three pieces in, three pieces out



```
GET / HTTP/1.1
Host: www.ucsd.edu
User-Agent: Thomas Super Browser
X-Demo: Yep

HTTP/1.1 200 OK
Date: Thu, 10 Oct 2013 22:09:10 GMT
Server: Apache/2.0.63 (Unix) DAV/2 mod_perl/2.0.4 Perl/v5.8.4
Last-Modified: Thu, 10 Oct 2013 21:35:31 GMT
ETag: "6ec2-c7f86ec0"
Accept-Ranges: bytes
Content-Length: 28354
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="en">

<head>
```

## HTTP Challenges

- Performance
  - Issue: Lots of requests
  - Solution: Request reduction via bundling and caching

Performance Golden Rule: Less data, less often and close by

## HTTP Challenges

- Security
  - Or lack there of
  - HTTPS only addresses about 1/3 of the problem

Security Golden Rule: Trust no user nor data

## HTTP Challenges

- State Management (since stateless)
- Possible Solutions

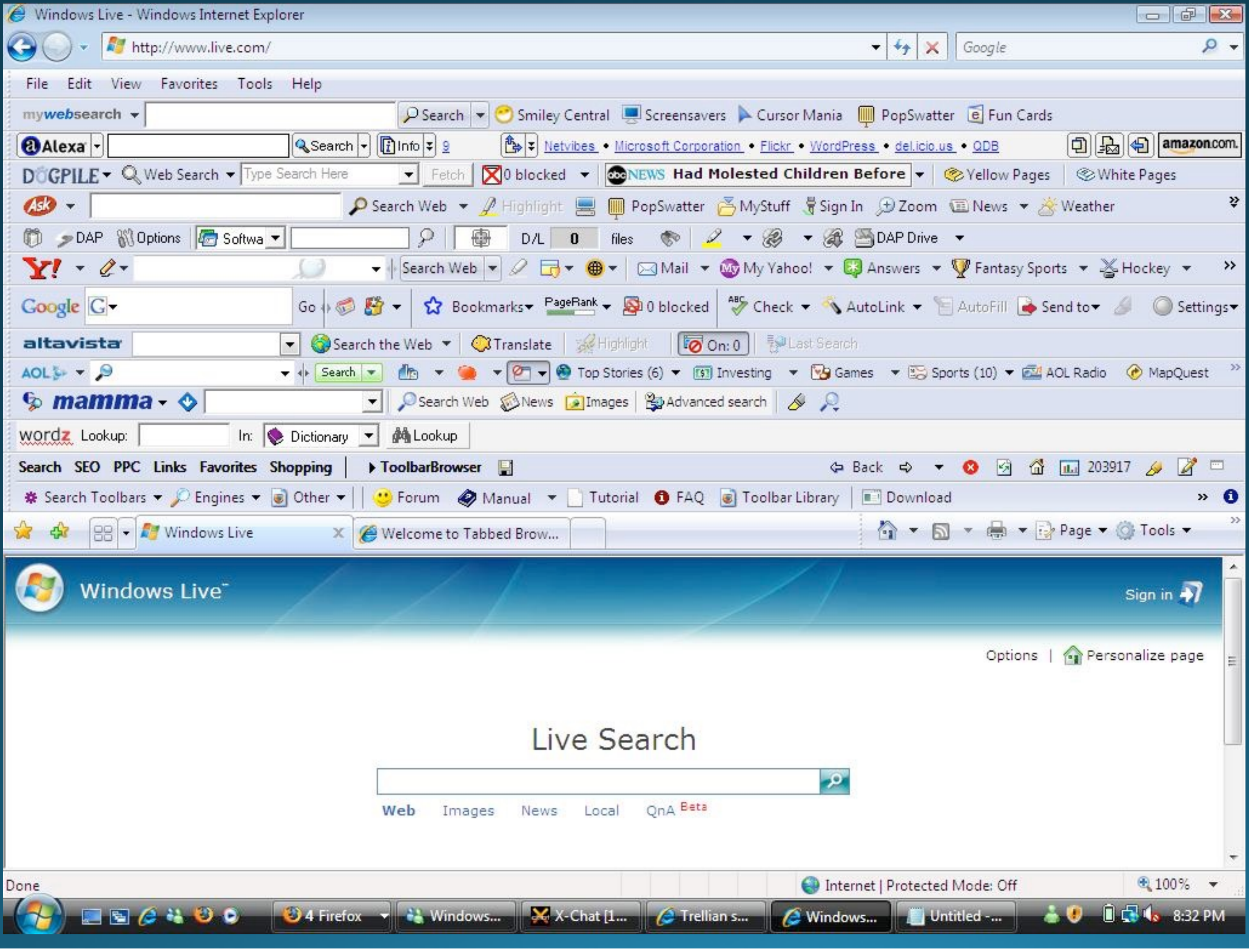
1. Dirty URLs
2. Hidden Form Fields
3. Cookies
4. Local Storage (please no!)
5. Move to stateful (ex. WebSocket)

- Privacy concerns undermine state issues
- Framework abstraction can really hurt us here if we don't understand mechanisms

## HTTP Challenges

- Protocol Misunderstanding is Widespread
  - Cave Man HTTP: Ugh, Bugha Bugha
  - GET = Ugh, POST = Bugha Bugha interchange at will
- We aim to build modern service oriented architecture we need a deeper understanding of HTTP than that!
- Careful though are you sure you have HTTP verb support end to end? I think that is assumed wrongly more than we might want to admit. Good engineers would verify a system before using it (in code not just in theory!)

# Browsers!


Windows Live - Windows Internet Explorer
http://www.live.com/
Google
File Edit View Favorites Tools Help
mywebsearch Search Smiley Central Screensavers Cursor Mania PopSwatter Fun Cards
Alexa Search Info Netvibes Microsoft Corporation Flickr WordPress del.icio.us QDB amazon.com
DOGPILE Web Search Type Search Here Fetch 0 blocked NEWS Had Molested Children Before Yellow Pages White Pages
Ask Search Web Highlight PopSwatter MyStuff Sign In Zoom News Weather
DAP Options Softwa D/L 0 files DAP Drive
Y! Search Web Mail My Yahoo! Answers Fantasy Sports Hockey
Google Go Bookmarks PageRank 0 blocked Check AutoLink AutoFill Send to Settings
altavista Search the Web Translate Highlight On: 0 Last Search
AOL Search Top Stories (6) Investing Games Sports (10) AOL Radio MapQuest
mamma Search Web News Images Advanced search
wordz Lookup: In: Dictionary Lookup
Search SEO PPC Links Favorites Shopping ToolbarBrowser Back 203917
Search Toolbars Engines Other Forum Manual Tutorial FAQ Toolbar Library Download
Windows Live Welcome to Tabbed Brow... Page Tools
Windows Live Sign in
Options | Personalize page
Live Search
Web Images News Local QnA Beta
Done Internet | Protected Mode: Off 100%
4 Firefox Windows... X-Chat [1... Trellian s... Windows... Untitled -... 8:32 PM

## Browser Families

1. Mozilla/Firefox (Gecko) - Netscape descendent
2. Google Chrome (initially WebKit now Blink)
3. Safari (Webkit)
4. Internet Explorer (Trident and new IE)
5. Opera (Presto, Blink)

http://en.wikipedia.org/wiki/Comparison_of_web_browser_engines

- If browser is the OS should we know how browsers work?

http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/

## Not Even Close

- Goal: Trying to cram 20 weeks of material on HTTP, server, HTML, JS, CSS, etc. into a few hours
- Key points still to come in parts 3 and 4 which get more and more practical
- Sad Truth: Projects are undone or overbudget because of huge misconceptions about tech, server, etc. It is always best to know what we speak of!

## Our Project Medium

- Web App
  - Underline word Web
  - What does that mean?
    - HTTP, HTML, Browsers, Servers, etc.

## HTML Points

- Traditional SGML Based
  - HTML 2, ~~3~~, 3.2, 4 (strict and transitional)
- XML Based
  - XHTML 1, 1.1
- Back to the future
  - HTML5 (no SGML DTD)
  - Odd Duck: XHTML5

## Doctypes

- Traditionally (X)HTML doctypes indicate dialect and DTD of the markup language in use

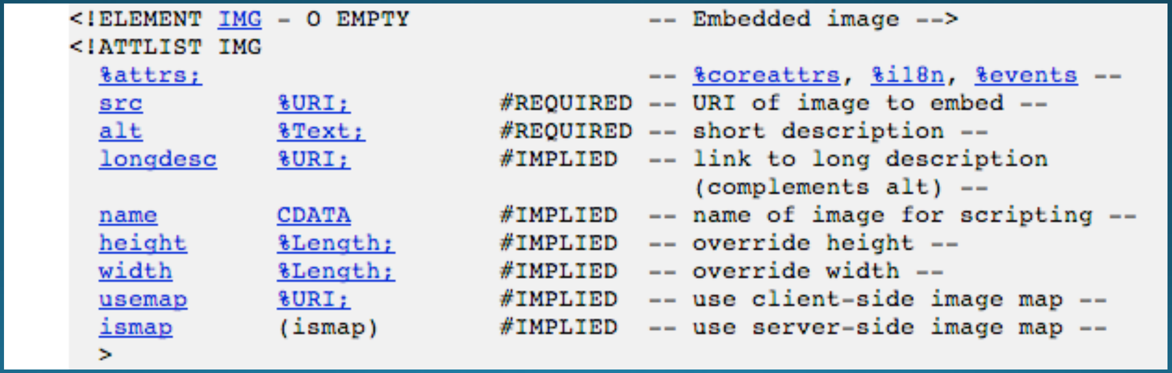
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

# What is a DTD?

- The actual reference (implicit) or linked DTD contains the rules of valid markup
- From http://www.w3.org/TR/REC-html40/sgml/dtd.html
- <img>

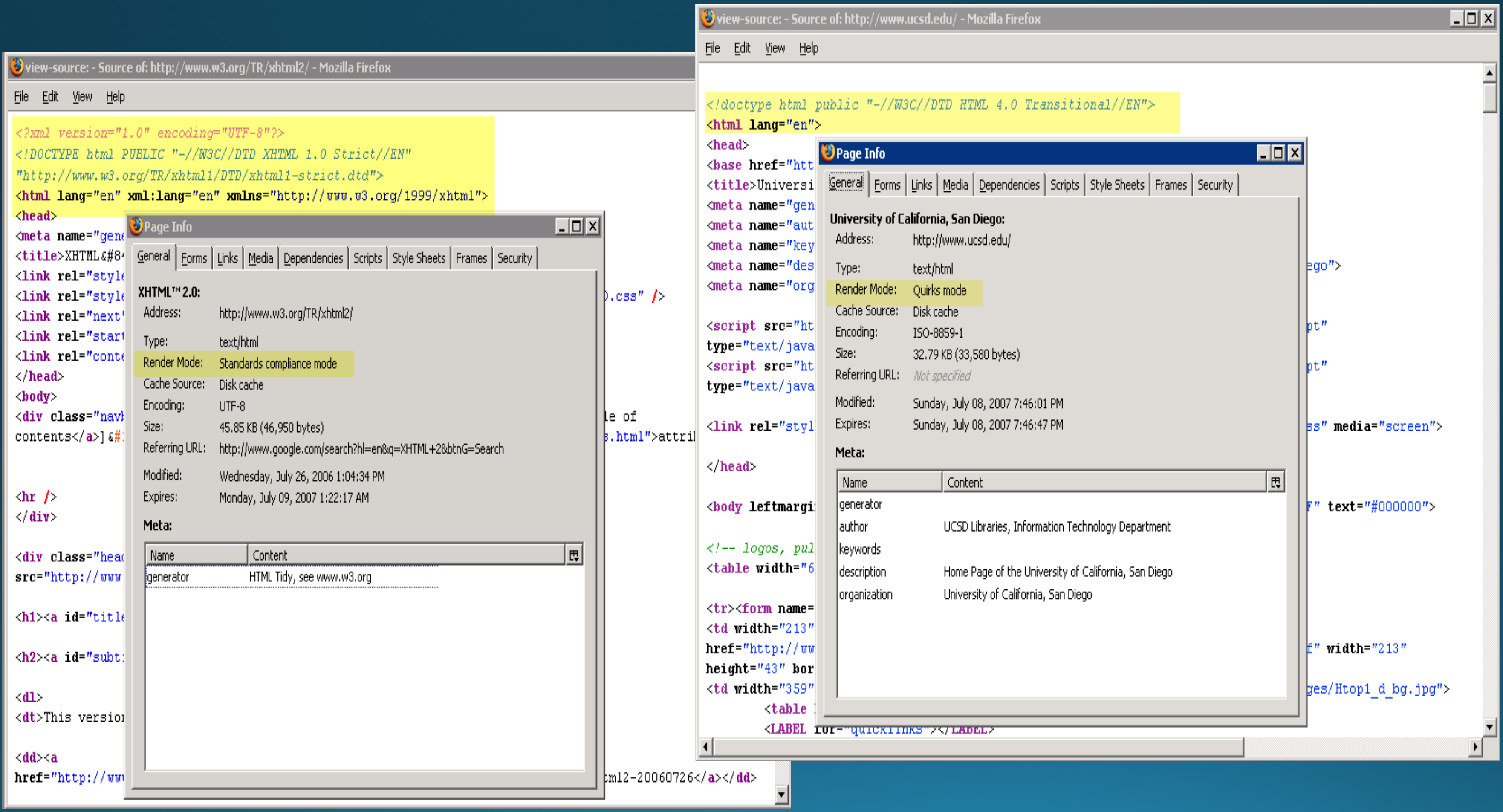


```
<!ELEMENT IMG - O EMPTY                        -- Embedded image -->
<!ATTLIST IMG
  %attrs;                                      -- %coreattrs, %i18n, %events --
  src         %URI;          #REQUIRED -- URI of image to embed --
  alt         %Text;         #REQUIRED -- short description --
  longdesc    %URI;          #IMPLIED  -- link to long description
                                          (complements alt) --
  name        CDATA          #IMPLIED  -- name of image for scripting --
  height      %Length;       #IMPLIED  -- override height --
  width       %Length;       #IMPLIED  -- override width --
  usemap      %URI;          #IMPLIED  -- use client-side image map --
  ismap       (ismap)        #IMPLIED  -- use server-side image map --
  >
```

## Uses of Doctypes

- Modern browsers are aware of the <!DOCTYPE> and will examine it to determine what rendering mode to enter (standards vs. quirk).
  - This process is often dubbed the "doctype switch"
- Using the <!DOCTYPE> declaration allows validation software to identify the DTD being followed in a document, and verify that the document is syntactically correct—in other words, that all tags used are part of a particular specification and are being used correctly.

# Doctype Switch in Action



view-source: - Source of: http://www.w3.org/TR/xhtml2/ - Mozilla Firefox

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="en" xml:lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
```

Page Info

**XHTML™ 2.0:**

| | |
|---|---|
| Address: | http://www.w3.org/TR/xhtml2/ |
| Type: | text/html |
| Render Mode: | Standards compliance mode |
| Cache Source: | Disk cache |
| Encoding: | UTF-8 |
| Size: | 45.85 KB (46,950 bytes) |
| Referring URL: | http://www.google.com/search?hl=en&q=XHTML+2&btnG=Search |
| Modified: | Wednesday, July 26, 2006 1:04:34 PM |
| Expires: | Monday, July 09, 2007 1:22:17 AM |

**Meta:**

| Name | Content |
|---|---|
| generator | HTML Tidy, see www.w3.org |

view-source: - Source of: http://www.ucsd.edu/ - Mozilla Firefox

```
<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="en">
<head>
```

Page Info

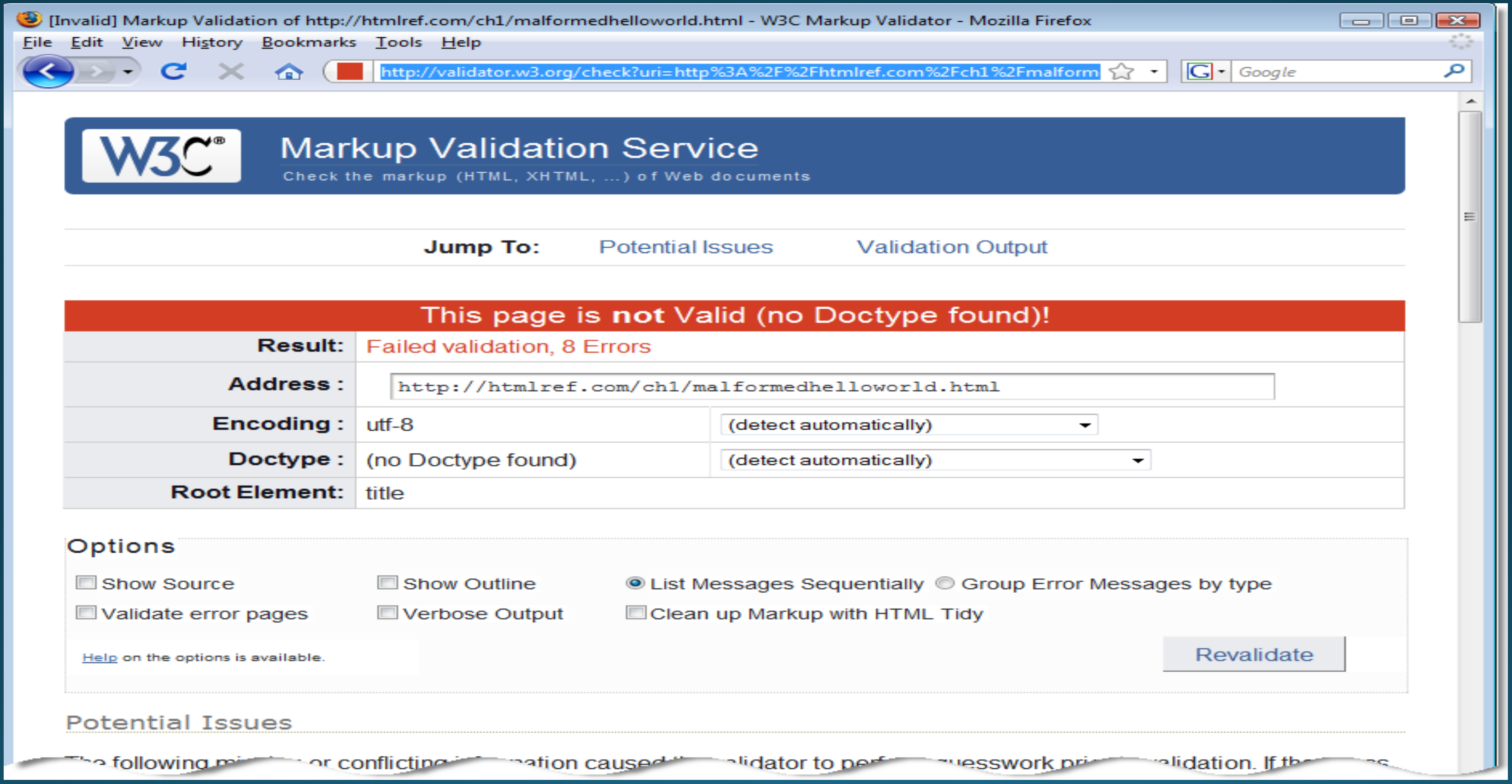
**University of California, San Diego:**

| | |
|---|---|
| Address: | http://www.ucsd.edu/ |
| Type: | text/html |
| Render Mode: | Quirks mode |
| Cache Source: | Disk cache |
| Encoding: | ISO-8859-1 |
| Size: | 32.79 KB (33,580 bytes) |
| Referring URL: | *Not specified* |
| Modified: | Sunday, July 08, 2007 7:46:01 PM |
| Expires: | Sunday, July 08, 2007 7:46:47 PM |

**Meta:**

| Name | Content |
|---|---|
| generator | |
| author | UCSD Libraries, Information Technology Department |
| keywords | |
| description | Home Page of the University of California, San Diego |
| organization | University of California, San Diego |

# Validation - Fail


[Invalid] Markup Validation of http://htmlref.com/ch1/malformedhelloworld.html - W3C Markup Validator - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://validator.w3.org/check?uri=http%3A%2F%2Fhtmlref.com%2Fch1%2Fmalform
Google
W3C
Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents
Jump To: Potential Issues Validation Output

This page is not Valid (no Doctype found)!

| | | |
|---|---|---|
| Result: | Failed validation, 8 Errors | |
| Address : | http://htmlref.com/ch1/malformedhelloworld.html | |
| Encoding : | utf-8 | (detect automatically) |
| Doctype : | (no Doctype found) | (detect automatically) |
| Root Element: | title | |

Options

- [ ] Show Source
- [ ] Show Outline
- ● List Messages Sequentially ○ Group Error Messages by type
- [ ] Validate error pages
- [ ] Verbose Output
- [ ] Clean up Markup with HTML Tidy

Help on the options is available.

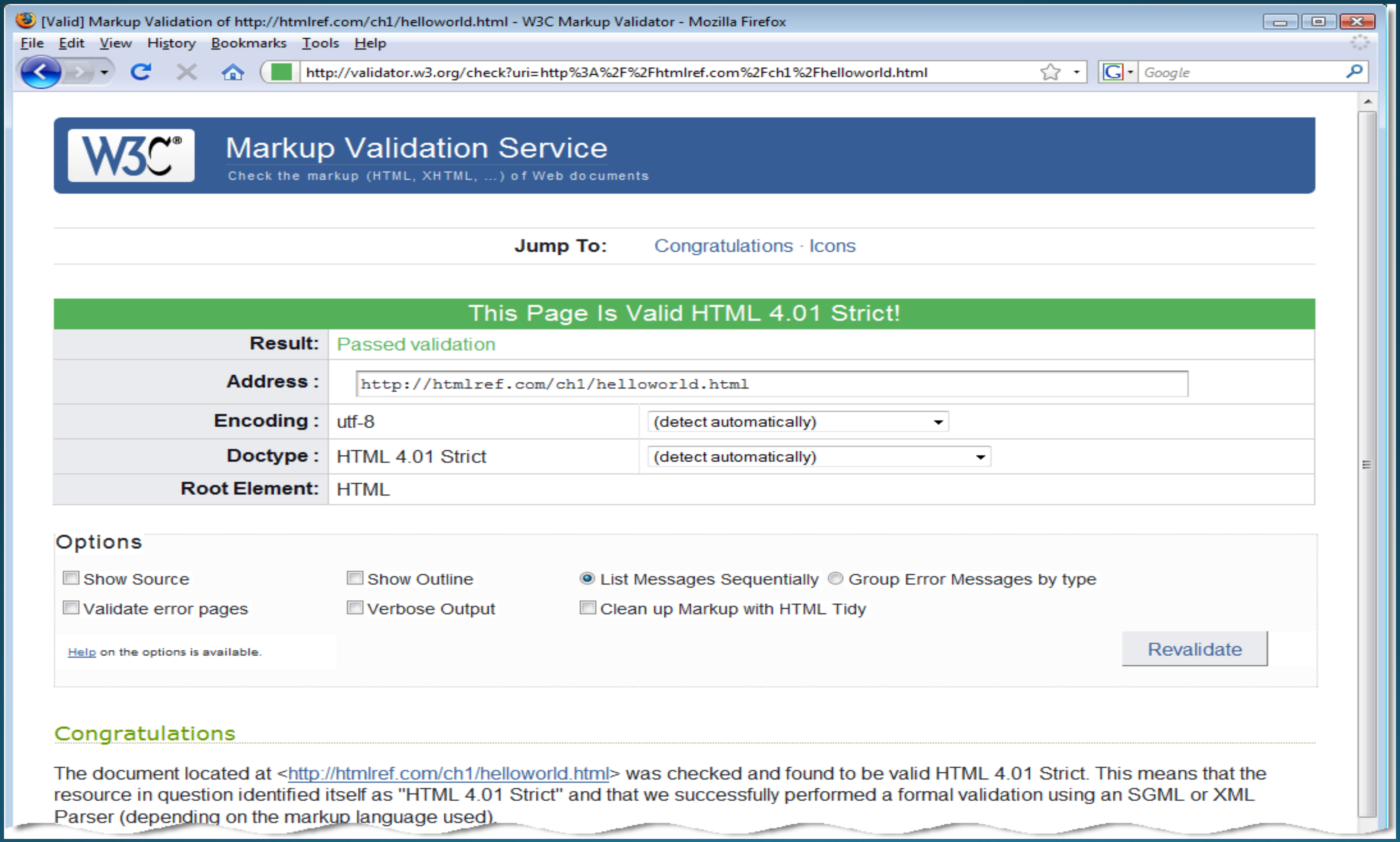
Potential Issues

Checking the conformance of a document to the stated (or inferred DTD)...do the tags meet the rules in the spec? Is it valid?

## Markup Characteristics

- well-formedness - following basic syntax rules for quotes, closing tags, case, etc.
- validity - adhering to the allowed vocabulary of tags and how they may be used
- “Purple is an eager flying cactus who loves a dog sadly perhaps.” is well formed English sentence but it isn’t valid*

## Validation - Pass


[Valid] Markup Validation of http://htmlref.com/ch1/helloworld.html - W3C Markup Validator - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://validator.w3.org/check?uri=http%3A%2F%2Fhtmlref.com%2Fch1%2Fhelloworld.html
Google

Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

**Jump To:** Congratulations · Icons

**This Page Is Valid HTML 4.01 Strict!**

| | | |
|---|---|---|
| **Result:** | Passed validation | |
| **Address :** | http://htmlref.com/ch1/helloworld.html | |
| **Encoding :** | utf-8 | (detect automatically) |
| **Doctype :** | HTML 4.01 Strict | (detect automatically) |
| **Root Element:** | HTML | |

Options

☐ Show Source ☐ Show Outline [✓ List Messages Sequentially] Group Error Messages by type
☐ Validate error pages ☐ Verbose Output ☐ Clean up Markup with HTML Tidy

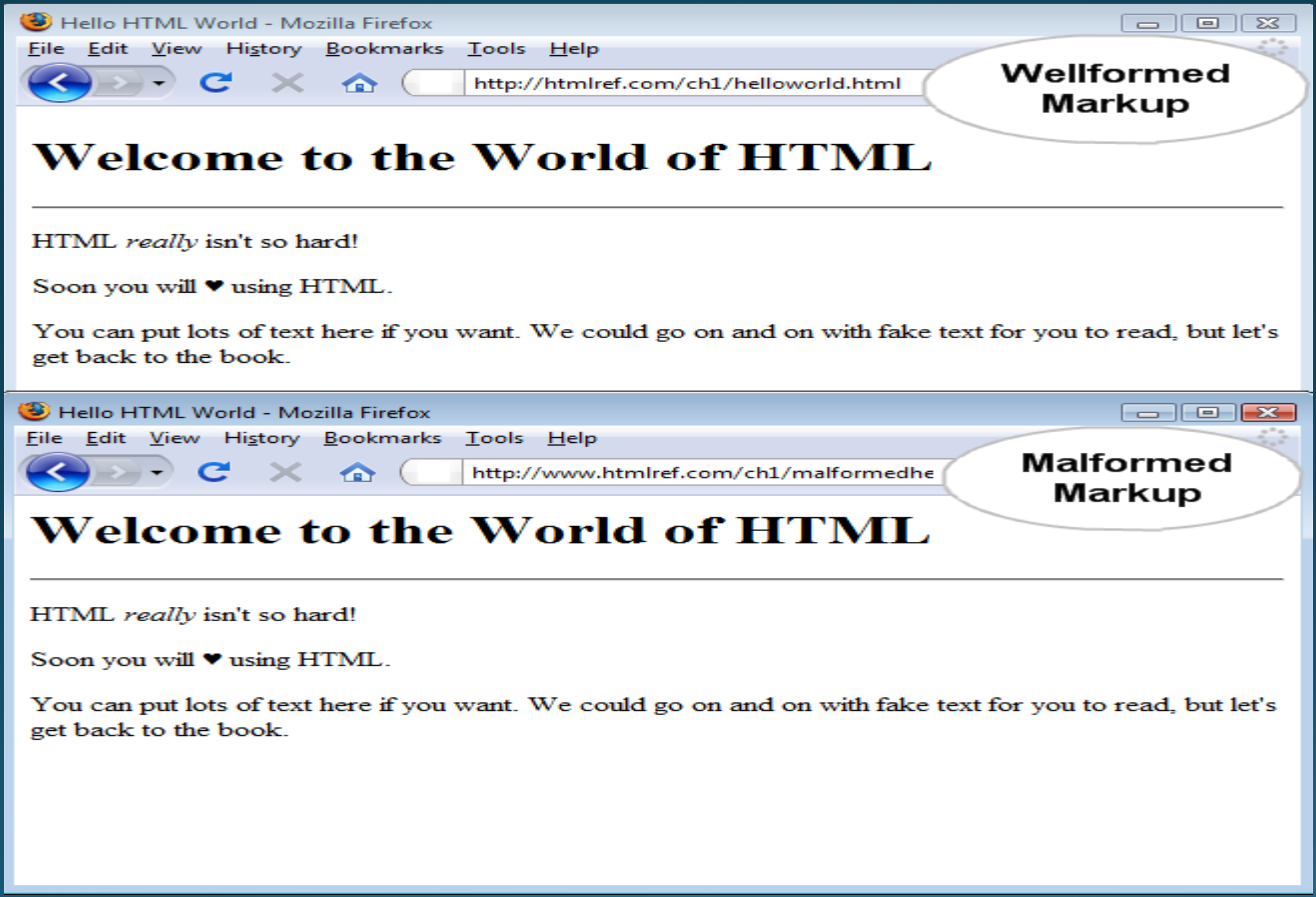
Help on the options is available.

Revalidate

Congratulations

The document located at <http://htmlref.com/ch1/helloworld.html> was checked and found to be valid HTML 4.01 Strict. This means that the resource in question identified itself as "HTML 4.01 Strict" and that we successfully performed a formal validation using an SGML or XML Parser (depending on the markup language used).

# Why Bother? Part 1


Hello HTML World - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://htmlref.com/ch1/helloworld.html
Wellformed Markup

Welcome to the World of HTML

HTML *really* isn't so hard!

Soon you will ❤ using HTML.

You can put lots of text here if you want. We could go on and on with fake text for you to read, but let's get back to the book.

Hello HTML World - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.htmlref.com/ch1/malformedhe
Malformed Markup

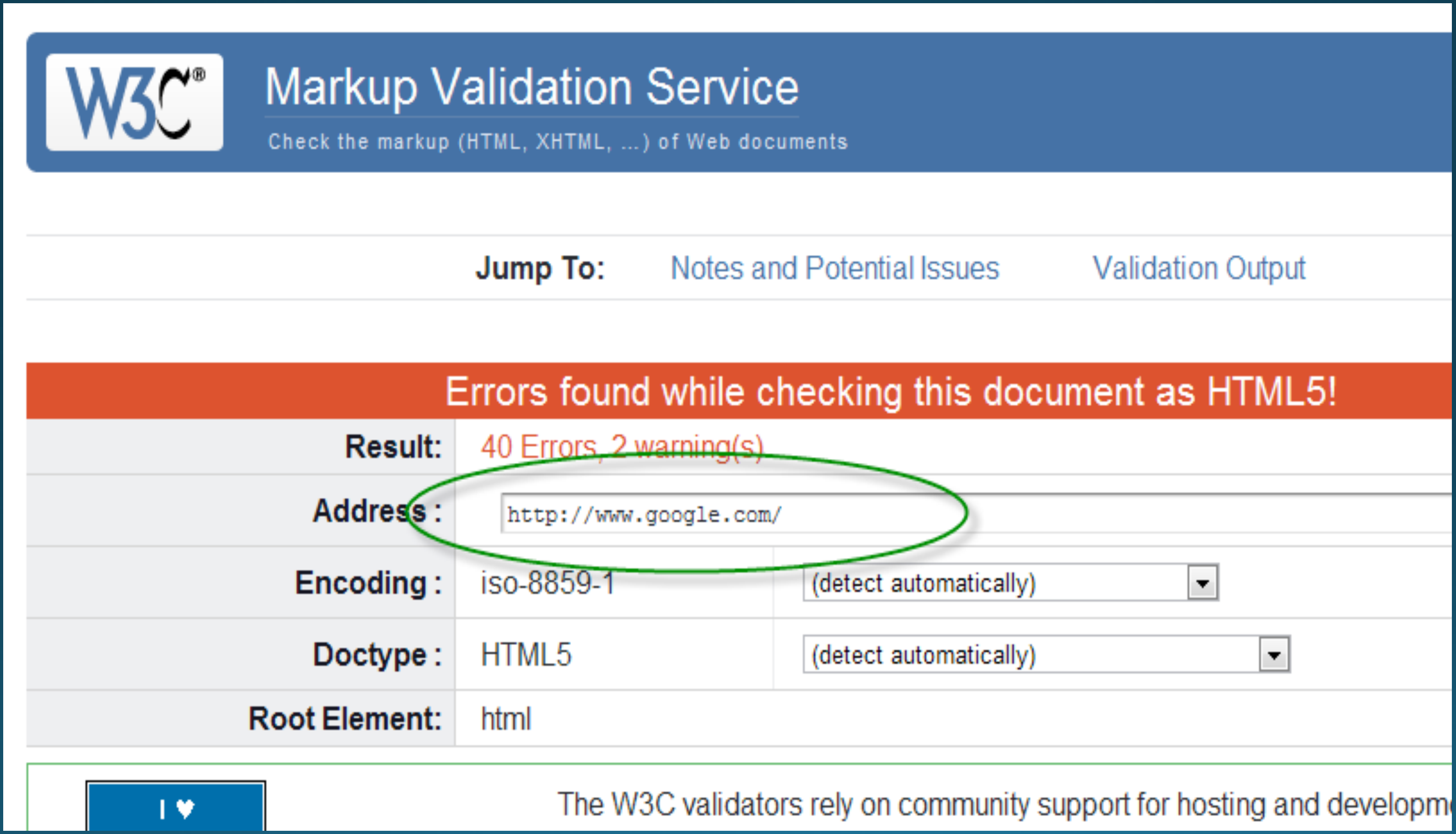
Welcome to the World of HTML

HTML *really* isn't so hard!

Soon you will ❤ using HTML.

You can put lots of text here if you want. We could go on and on with fake text for you to read, but let's get back to the book.

# Why Bother Part 2



Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: Notes and Potential Issues Validation Output

Errors found while checking this document as HTML5!

| | | |
|---|---|---|
| Result: | 40 Errors, 2 warning(s) | |
| Address : | http://www.google.com/ | |
| Encoding : | iso-8859-1 | (detect automatically) |
| Doctype : | HTML5 | (detect automatically) |
| Root Element: | html | |

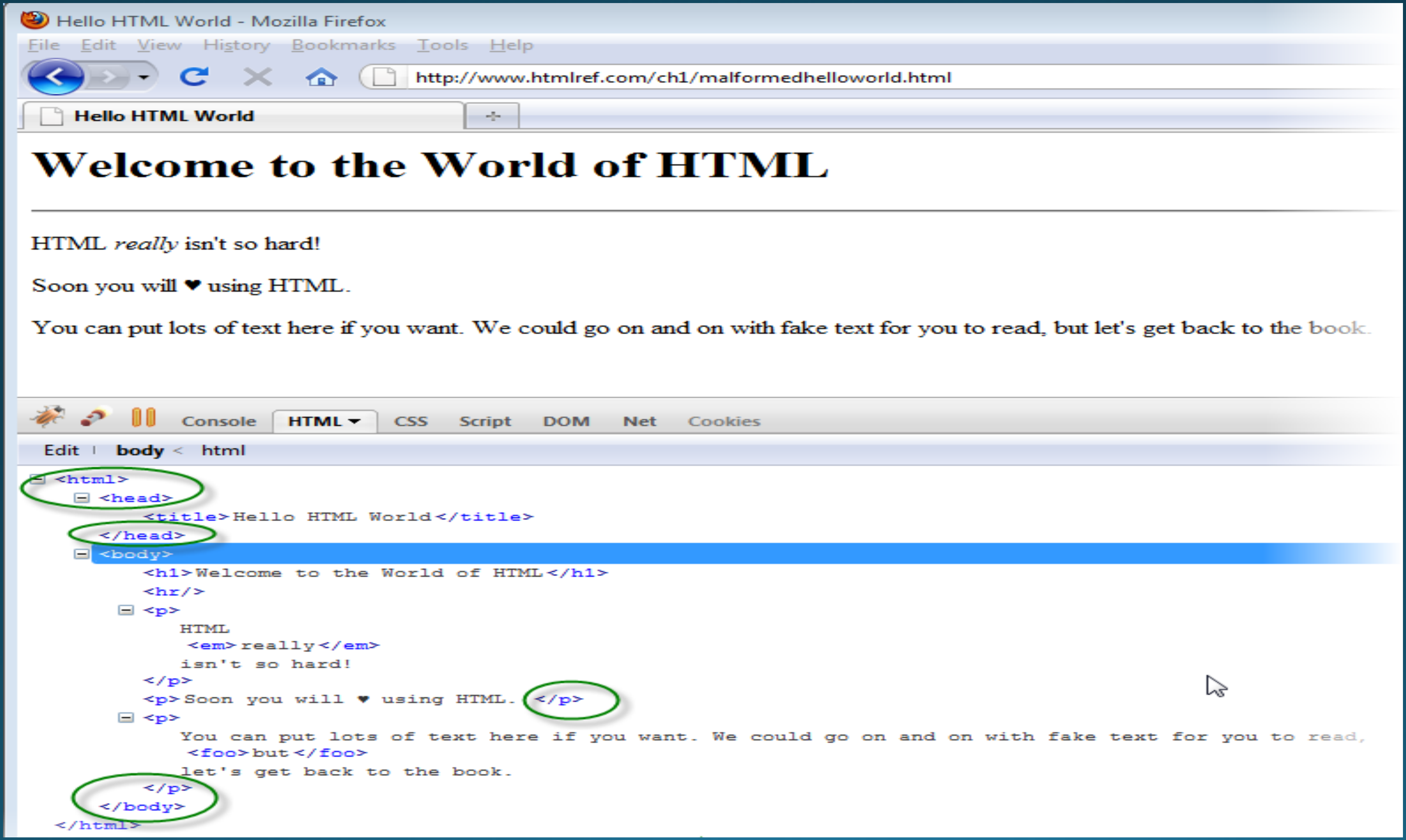
The W3C validators rely on community support for hosting and developm

## Tag Soup

Bad Example: http://htmlref.com/ch1/malformedhelloworld.html

```
<TITLE>Hello HTML World</title>
<!-- Simple hello malformed world -- example -->
</head>
<body>
<h1>Welcome to the World of HTML</H1>
<hr />
<p>HTML <eM>really</Em> isn't so hard!
<P>Soon you will &hearts; using HTML.
<p>You can put lots of text here if you want.
We could go on and on with fake text for you
to read, <foo>but</foo> let's get back to the book.
</html>
```

## Yum..fixed your soup


Hello HTML World - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.htmlref.com/ch1/malformedhelloworld.html
Hello HTML World
Console HTML CSS Script DOM Net Cookies
Edit body < html

Welcome to the World of HTML

HTML *really* isn't so hard!

Soon you will ♥ using HTML.

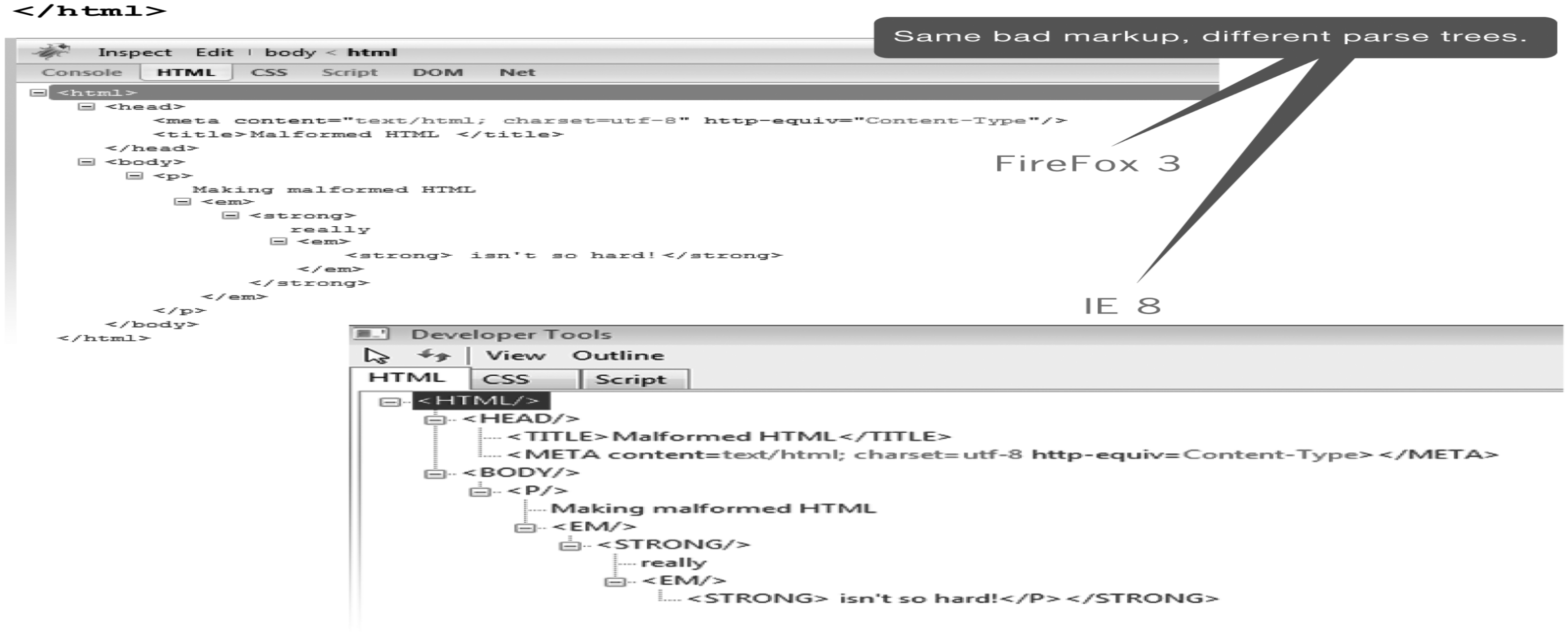
You can put lots of text here if you want. We could go on and on with fake text for you to read, but let's get back to the book.

```
<html>
  <head>
    <title>Hello HTML World</title>
  </head>
  <body>
    <h1>Welcome to the World of HTML</h1>
    <hr/>
    <p>
      HTML
      <em>really</em>
      isn't so hard!
    </p>
    <p>Soon you will ♥ using HTML.</p>
    <p>
      You can put lots of text here if you want. We could go on and on with fake text for you to read,
      <foo>but</foo>
      let's get back to the book.
    </p>
  </body>
</html>
```

# Guessing the parse

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8">
<title>Malformed HTML</title>
</head>
<body>
<p>Making malformed HTML <em><strong>really<em><strong>
isn't so hard!</p>
</body>
</html>
```

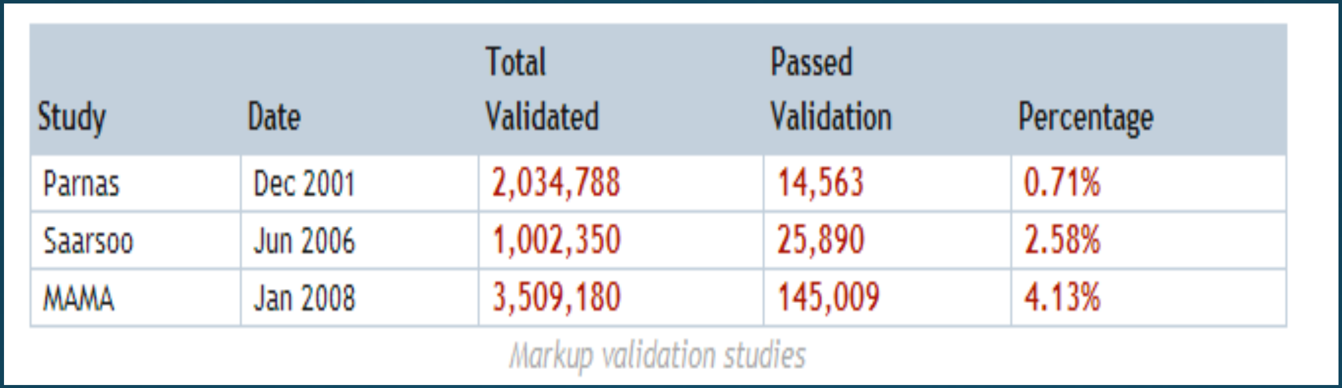


Same bad markup, different parse trees.

```
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
    <title>Malformed HTML </title>
  </head>
  <body>
    <p>
      Making malformed HTML
      <em>
        <strong>
          really
          <em>
            <strong> isn't so hard! </strong>
          </em>
        </strong>
      </em>
    </p>
  </body>
</html>
```

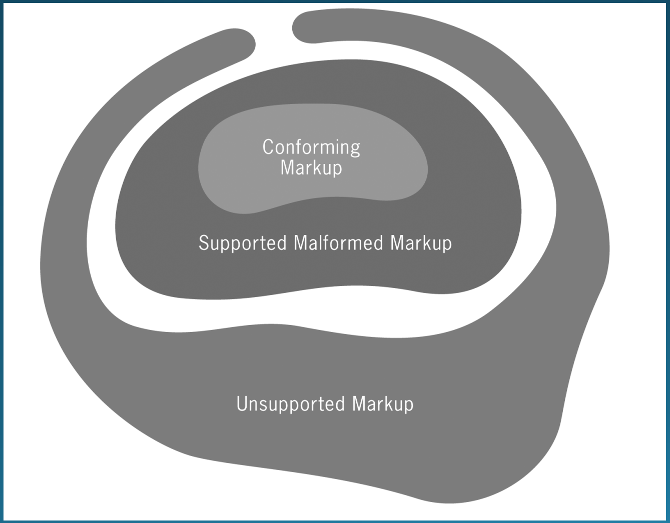
```
<HTML/>
  <HEAD/>
    <TITLE>Malformed HTML</TITLE>
    <META content=text/html; charset=utf-8 http-equiv=Content-Type></META>
  <BODY/>
    <P/>
      Making malformed HTML
      <EM/>
        <STRONG/>
          really
          <EM/>
            <STRONG> isn't so hard!</P></STRONG>
```

# Yeah everybody does it

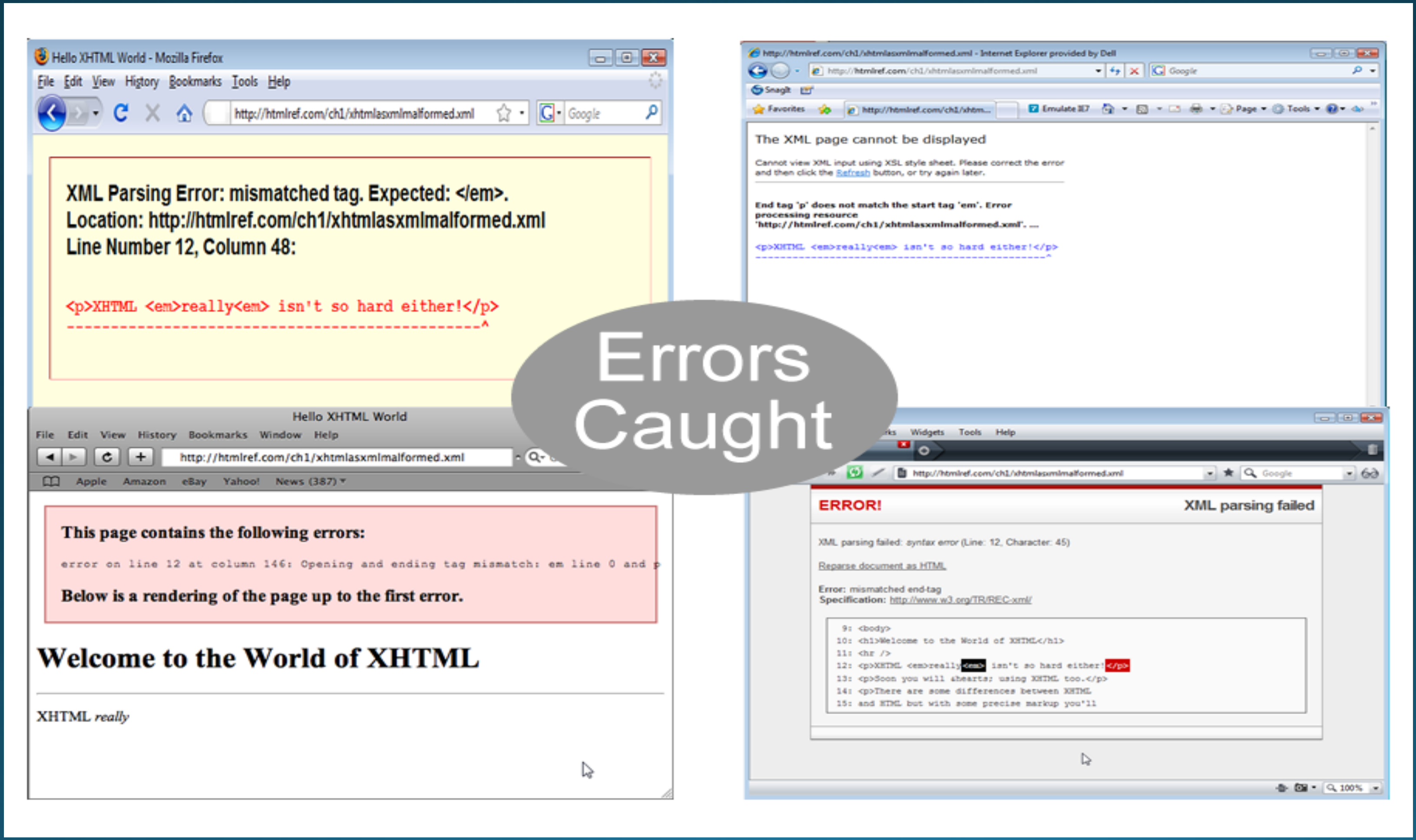


| Study | Date | Total Validated | Passed Validation | Percentage |
|---|---|---|---|---|
| Parnas | Dec 2001 | 2,034,788 | 14,563 | 0.71% |
| Saarsoo | Jun 2006 | 1,002,350 | 25,890 | 2.58% |
| MAMA | Jan 2008 | 3,509,180 | 145,009 | 4.13% |

*Markup validation studies*



## Enter XML / XHTML



Hello XHTML World - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://htmlref.com/ch1/xhtmlasxmlmalformed.xml
Errors Caught

XML Parsing Error: mismatched tag. Expected: </em>.
Location: http://htmlref.com/ch1/xhtmlasxmlmalformed.xml
Line Number 12, Column 48:

```
<p>XHTML <em>really<em> isn't so hard either!</p>
-----------------------------------------------^
```

The XML page cannot be displayed

Cannot view XML input using XSL style sheet. Please correct the error and then click the Refresh button, or try again later.

**End tag 'p' does not match the start tag 'em'. Error processing resource 'http://htmlref.com/ch1/xhtmlasxmlmalformed.xml'. ...**

```
<p>XHTML <em>really<em> isn't so hard either!</p>
-------------------------------------------------^
```

**This page contains the following errors:**

```
error on line 12 at column 146: Opening and ending tag mismatch: em line 0 and p
```

**Below is a rendering of the page up to the first error.**

**Welcome to the World of XHTML**

XHTML *really*

**ERROR!** **XML parsing failed**

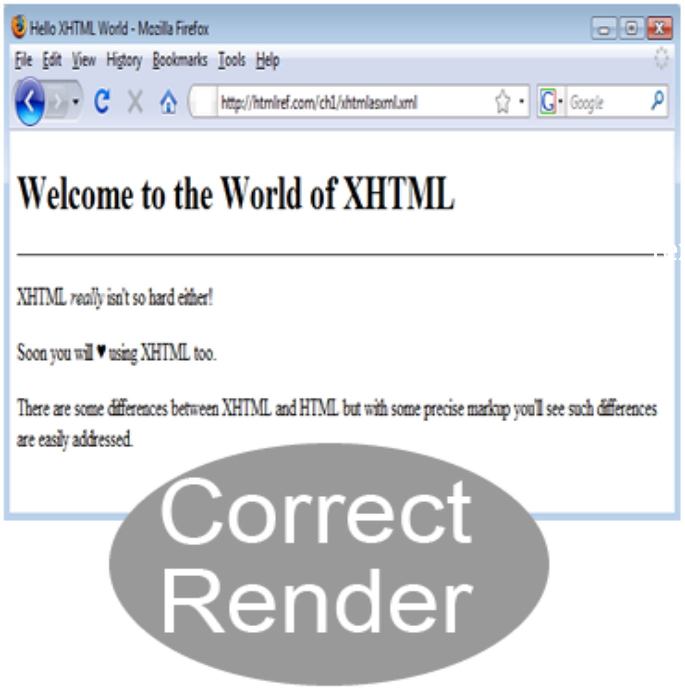
XML parsing failed: syntax error (Line: 12, Character: 45)

Reparse document as HTML

**Error:** mismatched end-tag
**Specification:** http://www.w3.org/TR/REC-xml/

```
 9: <body>
10: <h1>Welcome to the World of XHTML</h1>
11: <hr />
12: <p>XHTML <em>really<em> isn't so hard either!</p>
13: <p>Soon you will &hearts; using XHTML too.</p>
14: <p>There are some differences between XHTML
15: and HTML but with some precise markup you'll
```

## Unfortunately…even if you wanted to

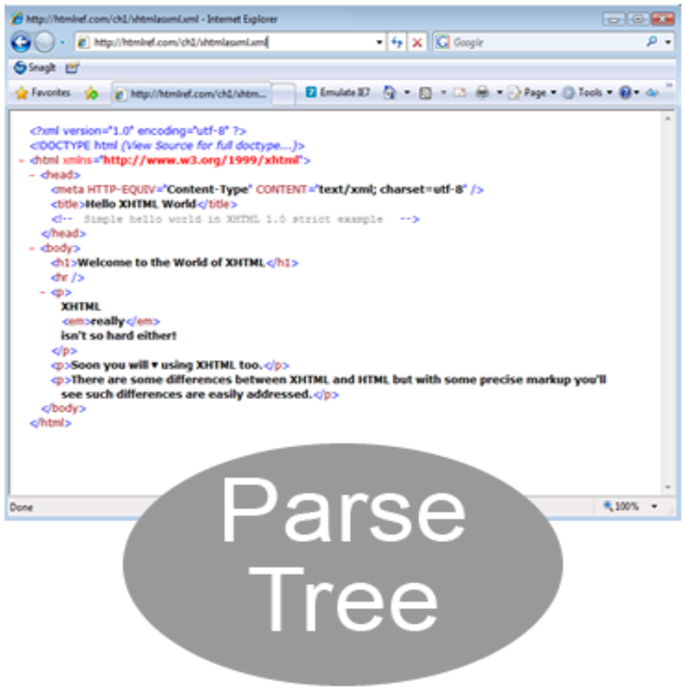

Hello XHTML World - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://htmlref.com/ch1/xhtmlasxml.xml
Google
Correct Render

# Welcome to the World of XHTML

XHTML *really* isn't so hard either!

Soon you will ♥ using XHTML too.

There are some differences between XHTML and HTML but with some precise markup you'll see such differences are easily addressed.



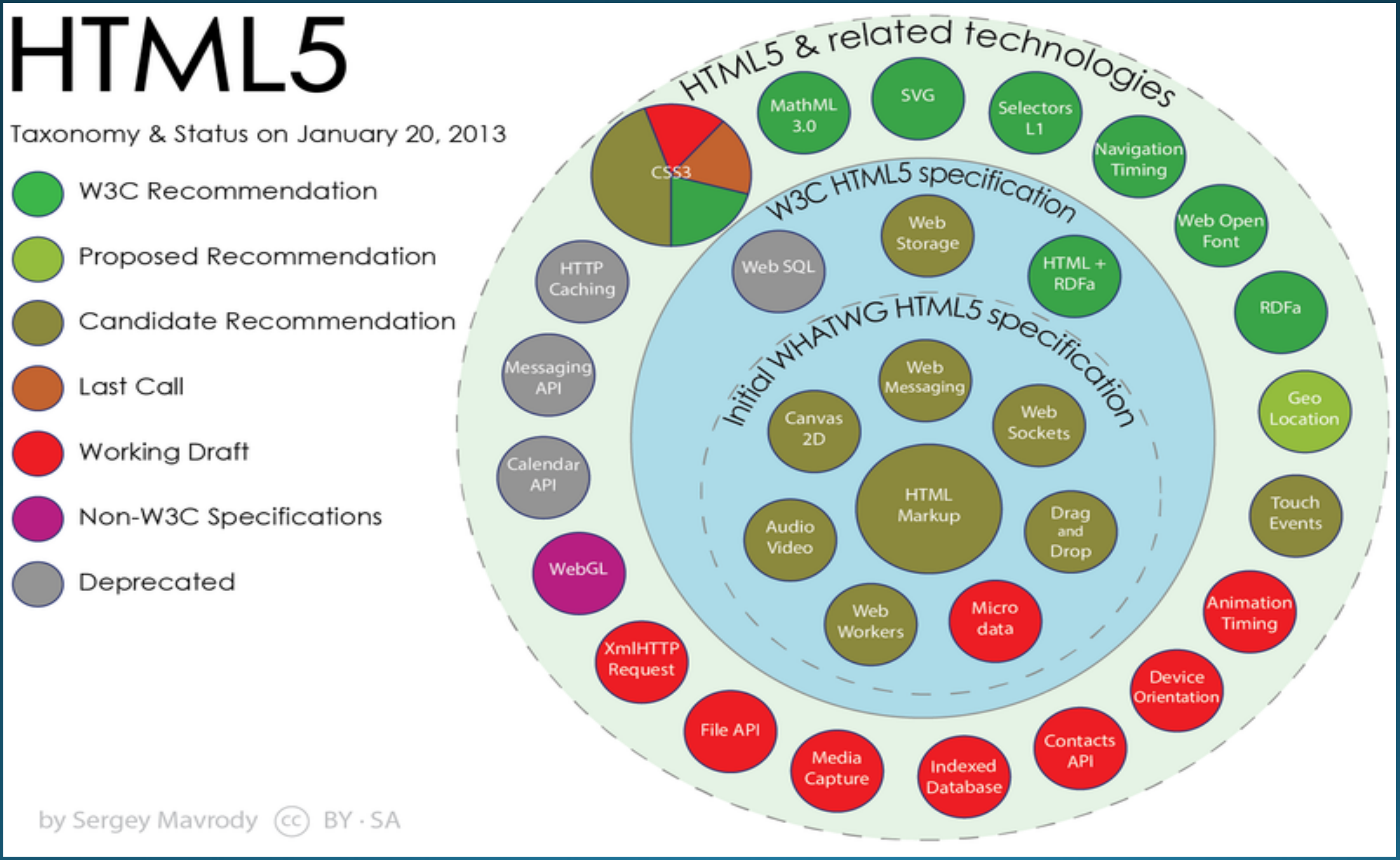
```
  <?xml version="1.0" encoding="utf-8" ?>
  <!DOCTYPE html (View Source for full doctype...)>
- <html xmlns="http://www.w3.org/1999/xhtml">
  - <head>
      <meta HTTP-EQUIV="Content-Type" CONTENT="text/xml; charset=utf-8" />
      <title>Hello XHTML World</title>
      <!--  Simple hello world in XHTML 1.0 strict example  -->
    </head>
  - <body>
      <h1>Welcome to the World of XHTML</h1>
      <hr />
    - <p>
        XHTML
        <em>really</em>
        isn't so hard either!
      </p>
      <p>Soon you will ♥ using XHTML too.</p>
      <p>There are some differences between XHTML and HTML but with some precise markup you'll
        see such differences are easily addressed.</p>
    </body>
  </html>
```

Good news everyone! IE9+ fixed this.

## Back to the Future

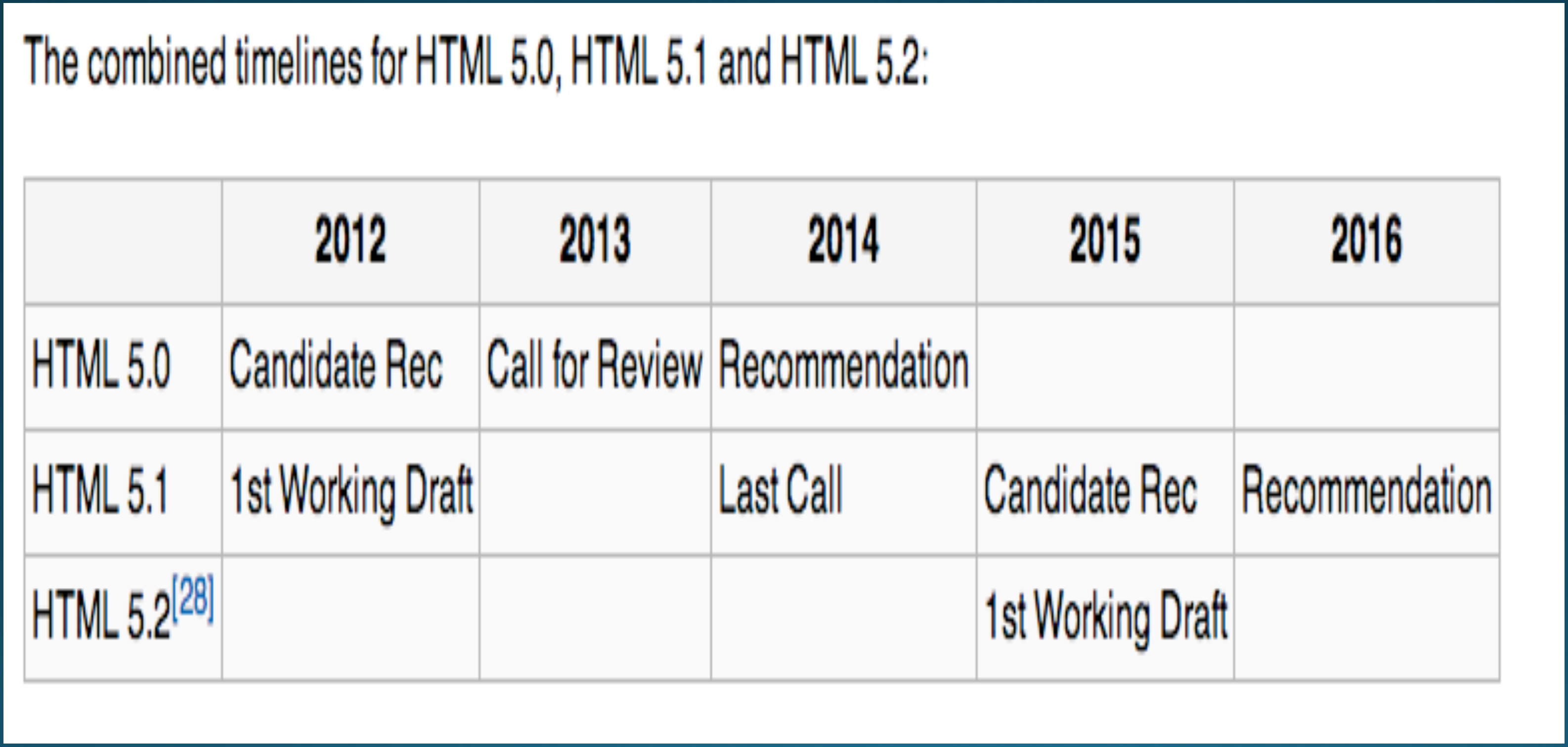
- But we aren't going there
- Too hard to get people to do it right, easier to get the browser vendors to have a malformedness handler agreement and return to the looseness
- HTML5 from a markup point of view is back to HTML4 with more new stuff…but worse

# HTML5 Viz Overview



## Spec Kitchen Sink

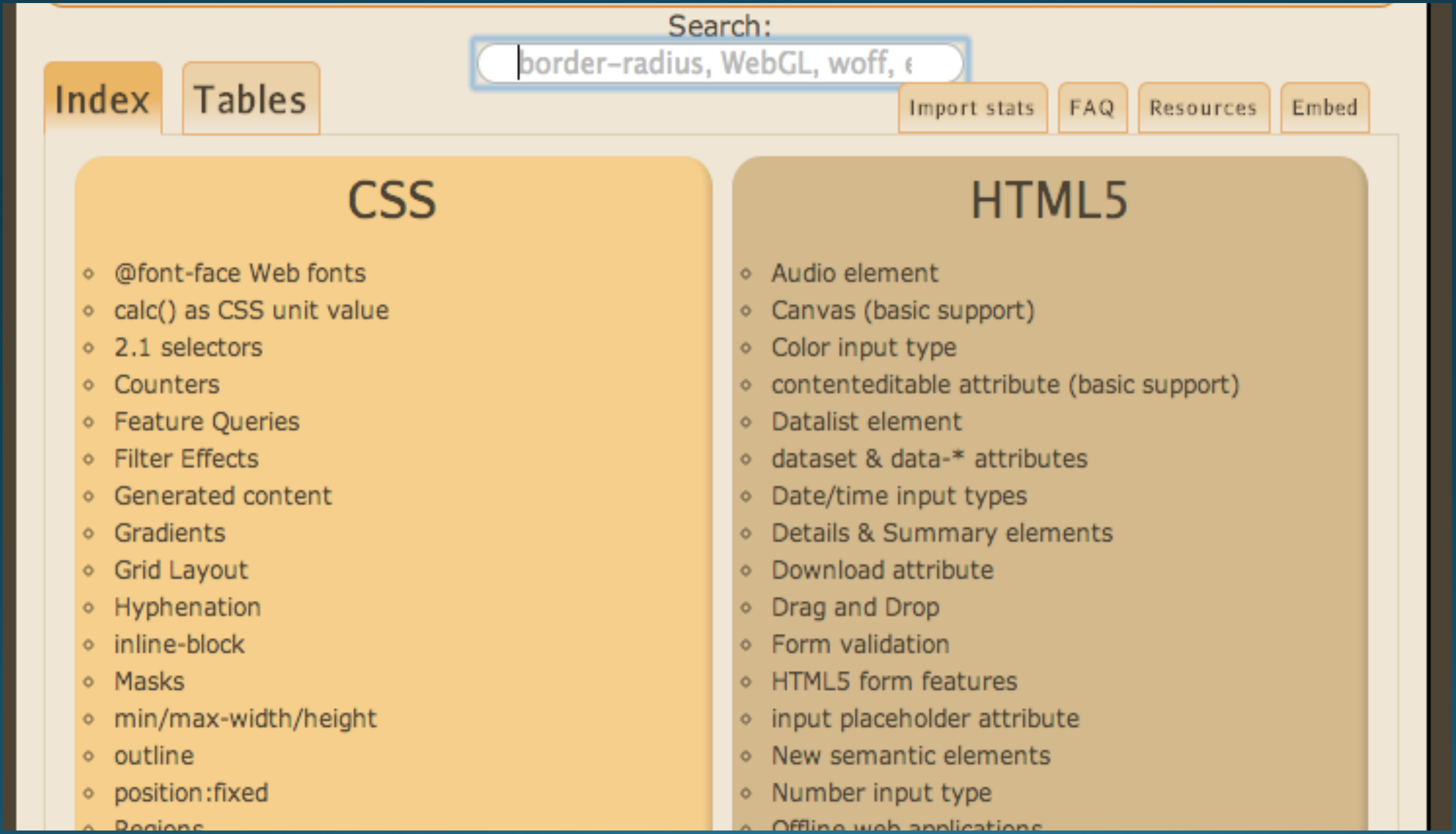
## Pffftt..HTML5? HTML5.2!



The combined timelines for HTML 5.0, HTML 5.1 and HTML 5.2:

| | 2012 | 2013 | 2014 | 2015 | 2016 |
|---|---|---|---|---|---|
| HTML 5.0 | Candidate Rec | Call for Review | Recommendation | | |
| HTML 5.1 | 1st Working Draft | | Last Call | Candidate Rec | Recommendation |
| HTML 5.2[28] | | | | 1st Working Draft | |

## Caniuse.com



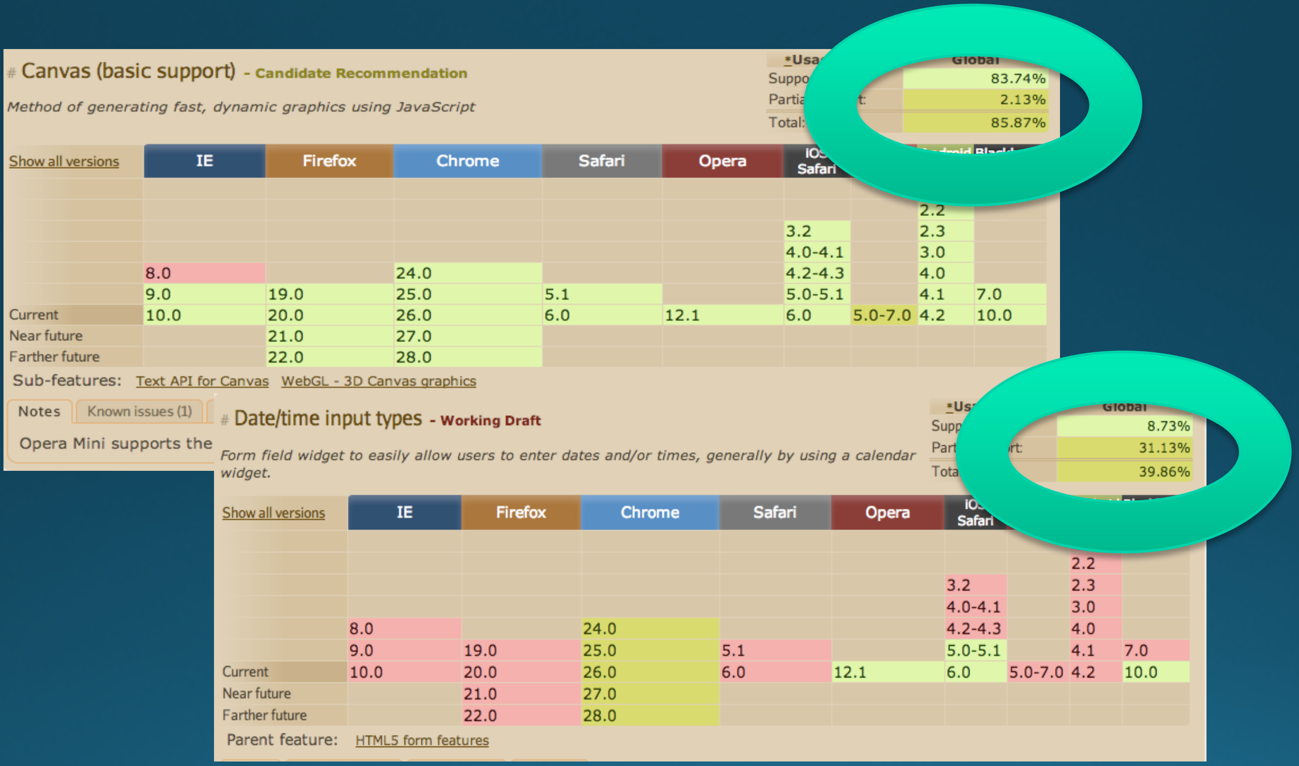
CSS

- @font-face Web fonts
- calc() as CSS unit value
- 2.1 selectors
- Counters
- Feature Queries
- Filter Effects
- Generated content
- Gradients
- Grid Layout
- Hyphenation
- inline-block
- Masks
- min/max-width/height
- outline
- position:fixed
- Regions

HTML5

- Audio element
- Canvas (basic support)
- Color input type
- contenteditable attribute (basic support)
- Datalist element
- dataset & data-* attributes
- Date/time input types
- Details & Summary elements
- Download attribute
- Drag and Drop
- Form validation
- HTML5 form features
- input placeholder attribute
- New semantic elements
- Number input type
- Offline web applications

# Proceed with caution?



# Canvas (basic support) - Candidate Recommendation

*Method of generating fast, dynamic graphics using JavaScript*

| *Usa | Global |
|---|---|
| Suppo | 83.74% |
| Partia t: | 2.13% |
| Total: | 85.87% |

| Show all versions | IE | Firefox | Chrome | Safari | Opera | iOS Safari | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| | | | | | | | | 2.2 | |
| | | | | | | 3.2 | | 2.3 | |
| | | | | | | 4.0-4.1 | | 3.0 | |
| | 8.0 | | 24.0 | | | 4.2-4.3 | | 4.0 | |
| | 9.0 | 19.0 | 25.0 | 5.1 | | 5.0-5.1 | | 4.1 | 7.0 |
| Current | 10.0 | 20.0 | 26.0 | 6.0 | 12.1 | 6.0 | 5.0-7.0 | 4.2 | 10.0 |
| Near future | | 21.0 | 27.0 | | | | | | |
| Farther future | | 22.0 | 28.0 | | | | | | |

Sub-features: Text API for Canvas WebGL - 3D Canvas graphics

Notes Known issues (1)

Opera Mini supports the

# Date/time input types - Working Draft

*Form field widget to easily allow users to enter dates and/or times, generally by using a calendar widget.*

| *Us | Global |
|---|---|
| Supp | 8.73% |
| Part rt: | 31.13% |
| Tota | 39.86% |

| Show all versions | IE | Firefox | Chrome | Safari | Opera | iOS Safari | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| | | | | | | | | 2.2 | |
| | | | | | | 3.2 | | 2.3 | |
| | | | | | | 4.0-4.1 | | 3.0 | |
| | 8.0 | | 24.0 | | | 4.2-4.3 | | 4.0 | |
| | 9.0 | 19.0 | 25.0 | 5.1 | | 5.0-5.1 | | 4.1 | 7.0 |
| Current | 10.0 | 20.0 | 26.0 | 6.0 | 12.1 | 6.0 | 5.0-7.0 | 4.2 | 10.0 |
| Near future | | 21.0 | 27.0 | | | | | | |
| Farther future | | 22.0 | 28.0 | | | | | | |

Parent feature: HTML5 form features

## Suggestions

- For public facing applications use HTML5 with an emphasis on realistic support
- Validate your markup, but understand when it may be appropriate to ignore validation
- Automate continuous validation on build and/or deploy
- Use XML based markup in a contained environments where failures are not tolerable
    - Aim beyond well-formedness to validity

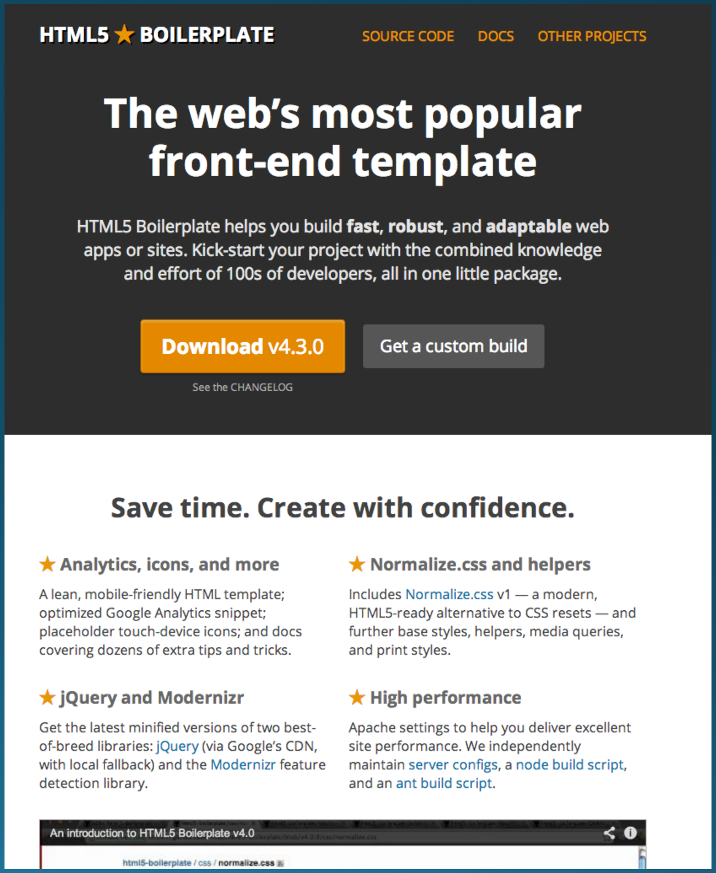
## Suggestions

- Adopt an HTML style guide
  - Casing, quotes " vs ', indent pattern, comments, attribute order, id and class name patterns, etc.
  - Employ a pretty printer for consistency
  - Use templates and scaffolds
    - Skeleton repo

## Suggestions - Boilerplates

Small Concerns

- Make sure you know what it is doing
- Are you including random stuff in a boilerplate bulking things up needlessly?



HTML5 ★ BOILERPLATE

SOURCE CODE DOCS OTHER PROJECTS

The web's most popular front-end template

HTML5 Boilerplate helps you build **fast**, **robust**, and **adaptable** web apps or sites. Kick-start your project with the combined knowledge and effort of 100s of developers, all in one little package.

Download v4.3.0 Get a custom build

See the CHANGELOG

Save time. Create with confidence.

★ Analytics, icons, and more

A lean, mobile-friendly HTML template; optimized Google Analytics snippet; placeholder touch-device icons; and docs covering dozens of extra tips and tricks.

★ Normalize.css and helpers

Includes Normalize.css v1 — a modern, HTML5-ready alternative to CSS resets — and further base styles, helpers, media queries, and print styles.

★ jQuery and Modernizr

Get the latest minified versions of two best-of-breed libraries: jQuery (via Google's CDN, with local fallback) and the Modernizr feature detection library.

★ High performance

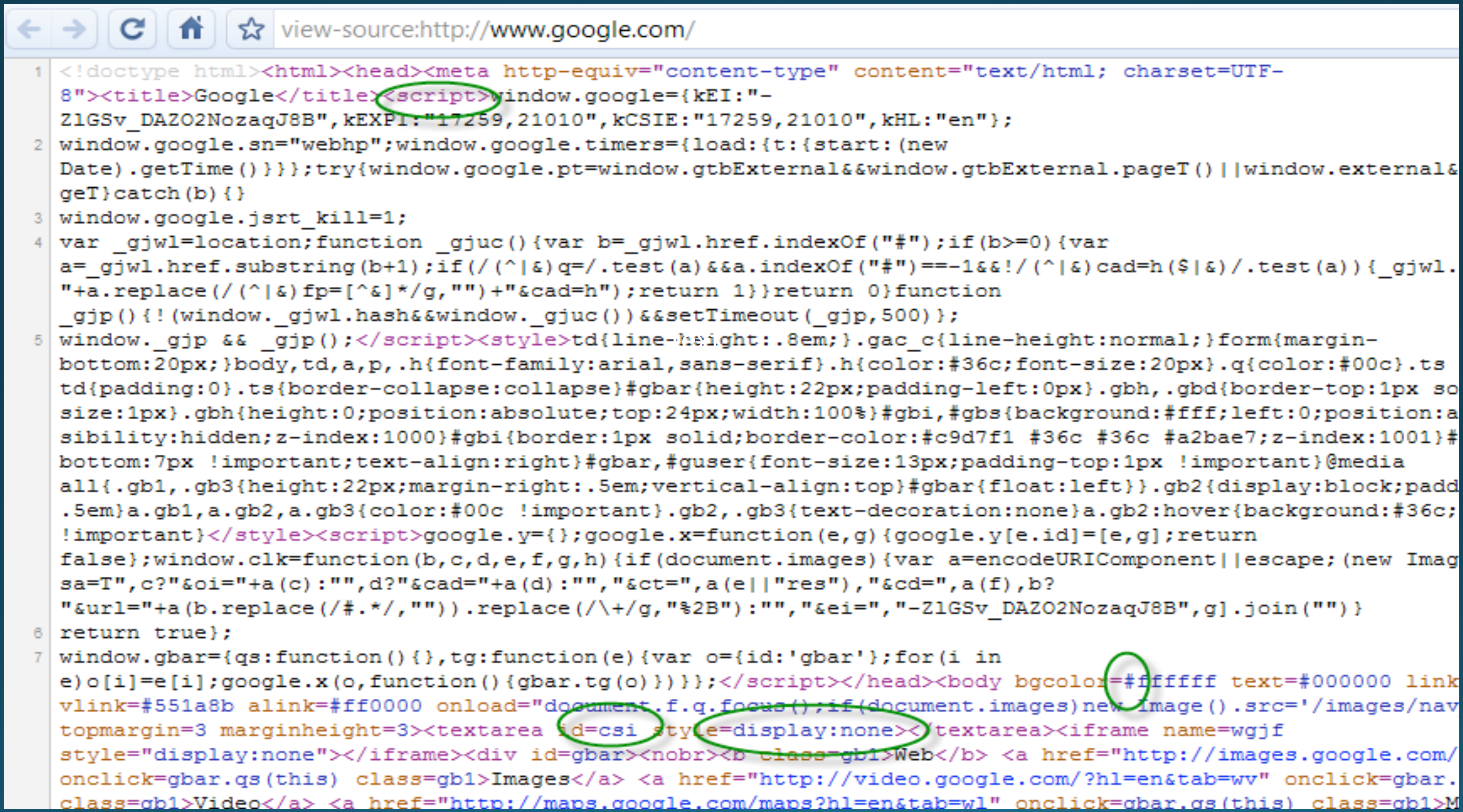
Apache settings to help you deliver excellent site performance. We independently maintain server configs, a node build script, and an ant build script.

An introduction to HTML5 Boilerplate v4.0

html5-boilerplate / css / normalize.css

http://html5boilerplate.com

# Suggestions - Minify



```
<!doctype html><html><head><meta http-equiv="content-type" content="text/html; charset=UTF-
8"><title>Google</title><script>window.google={kEI:"-
Z1GSv_DAZO2NozaqJ8B",kEXPI:"17259,21010",kCSIE:"17259,21010",kHL:"en"};
window.google.sn="webhp";window.google.timers={load:{t:{start:(new
Date).getTime()}}};try{window.google.pt=window.gtbExternal&&window.gtbExternal.pageT()||window.external&
geT}catch(b){}
window.google.jsrt_kill=1;
var _gjwl=location;function _gjuc(){var b=_gjwl.href.indexOf("#");if(b>=0){var
a=_gjwl.href.substring(b+1);if(/(^|&)q=/.test(a)&&a.indexOf("#")==-1&&!/(^|&)cad=h($|&)/.test(a)){_gjwl.
"+a.replace(/(^|&)fp=[^&]*/g,"")+"&cad=h");return 1}}return 0}function
_gjp(){!(window._gjwl.hash&&window._gjuc())&&setTimeout(_gjp,500)};
window._gjp && _gjp();</script><style>td{line-height:.8em;}.gac_c{line-height:normal;}form{margin-
bottom:20px;}body,td,a,p,.h{font-family:arial,sans-serif}.h{color:#36c;font-size:20px}.q{color:#00c}.ts
td{padding:0}.ts{border-collapse:collapse}#gbar{height:22px;padding-left:0px}.gbh,.gbd{border-top:1px so
size:1px}.gbh{height:0;position:absolute;top:24px;width:100%}#gbi,#gbs{background:#fff;left:0;position:a
sibility:hidden;z-index:1000}#gbi{border:1px solid;border-color:#c9d7f1 #36c #36c #a2bae7;z-index:1001}#
bottom:7px !important;text-align:right}#gbar,#guser{font-size:13px;padding-top:1px !important}@media
all{.gb1,.gb3{height:22px;margin-right:.5em;vertical-align:top}#gbar{float:left}}.gb2{display:block;padd
.5em}a.gb1,a.gb2,a.gb3{color:#00c !important}.gb2,.gb3{text-decoration:none}a.gb2:hover{background:#36c;
!important}</style><script>google.y={};google.x=function(e,g){google.y[e.id]=[e,g];return
false};window.clk=function(b,c,d,e,f,g,h){if(document.images){var a=encodeURIComponent||escape;(new Imag
sa=T",c?"&oi="+a(c):"",d?"&cad="+a(d):"","&ct=",a(e||"res"),"&cd=",a(f),b?
"&url="+a(b.replace(/#.*/,"")).replace(/\+/g,"%2B"):"","&ei=","-Z1GSv_DAZO2NozaqJ8B",g].join("")}
return true};
window.gbar={qs:function(){},tg:function(e){var o={id:'gbar'};for(i in
e)o[i]=e[i];google.x(o,function(){gbar.tg(o)})}};</script></head><body bgcolor=#ffffff text=#000000 link
vlink=#551a8b alink=#ff0000 onload="document.f.q.focus();if(document.images)new Image().src='/images/nav
topmargin=3 marginheight=3><textarea id=csi style=display:none></textarea><iframe name=wgjf
style="display:none"></iframe><div id=gbar><nobr><b class=gb1>Web</b> <a href="http://images.google.com/
onclick=gbar.qs(this) class=gb1>Images</a> <a href="http://video.google.com/?hl=en&tab=wv" onclick=gbar.
class=gb1>Video</a> <a href="http://maps.google.com/maps?hl=en&tab=wl" onclick=gbar.qs(this) class=gb1>M
```

Mistakes on purpose … theoretical adherence vs reality

# Suggestion - Semantics First

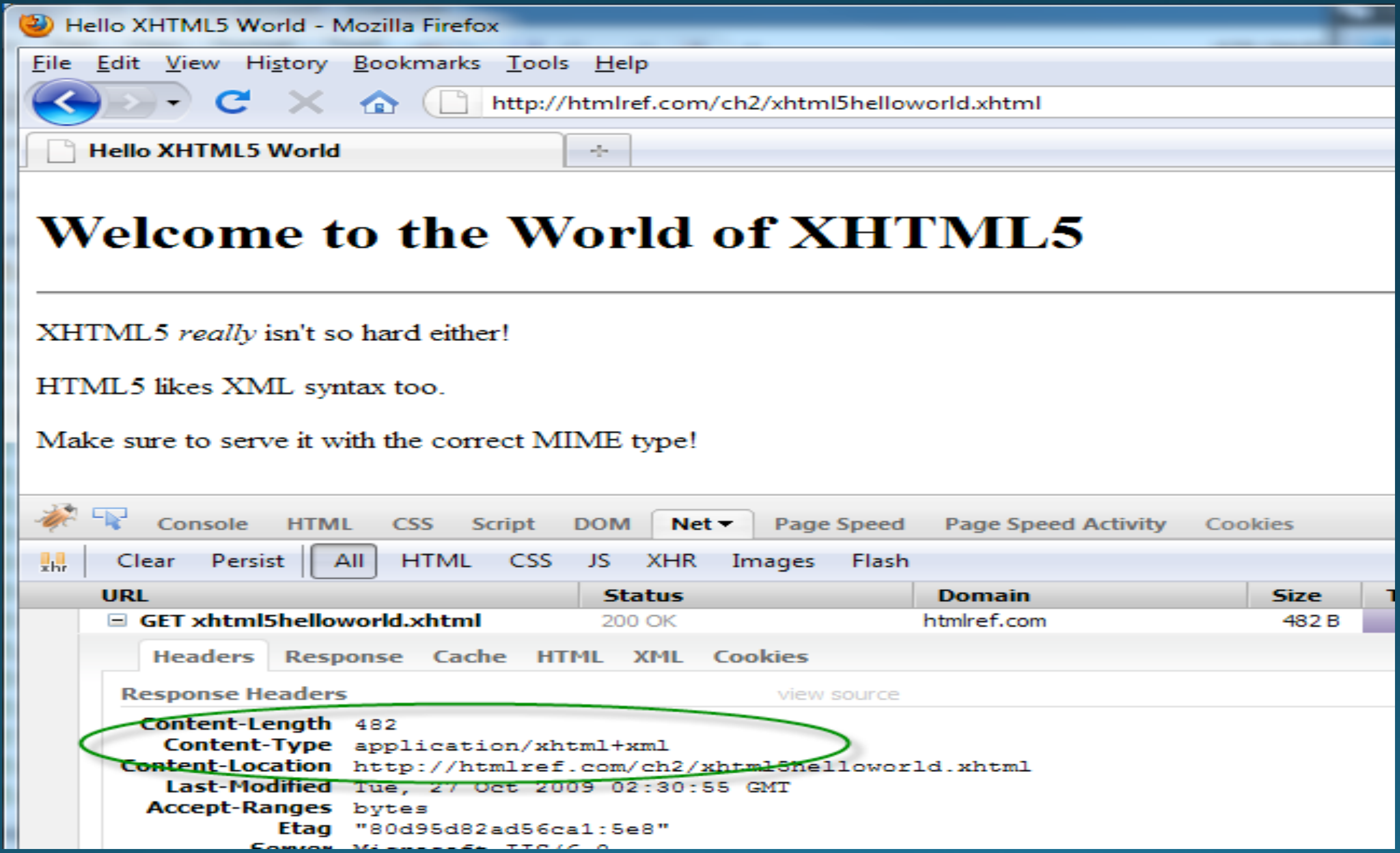
- Don’t catch <div>-itis
  - <div id=‘container’><div class=‘inner’>Blah…</div></div>
- Instead more meaning based tags
  - <header>, <footer>, <section>
- Avoid presentational markup (and thinking)
  - <font>, <b>, bgcolor, etc.
  - Assuming <h1> makes it big!
- Loosely couple style
  - <time style=‘background: orange’>Oct 31</time> <!-- so so --
    >
    <time class=”halloween”>Oct 31</time> <!-- better -->

## MIME

- Multipurpose Internet Mail Extensions
- Used by Web browsers via the HTTP Content-Type header
  - Upload - Post encoding commonly application/x-www-form-urlencoded
  - Download - text/html, image/jpg, etc.
  - Determine what to do in browser*

* most of the time

## Get the MIME Right


Hello XHTML5 World - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://htmlref.com/ch2/xhtml5helloworld.xhtml
Hello XHTML5 World
Console HTML CSS Script DOM Net Page Speed Page Speed Activity Cookies
Clear Persist All HTML CSS JS XHR Images Flash
Headers Response Cache HTML XML Cookies
view source

# Welcome to the World of XHTML5

XHTML5 *really* isn't so hard either!

HTML5 likes XML syntax too.

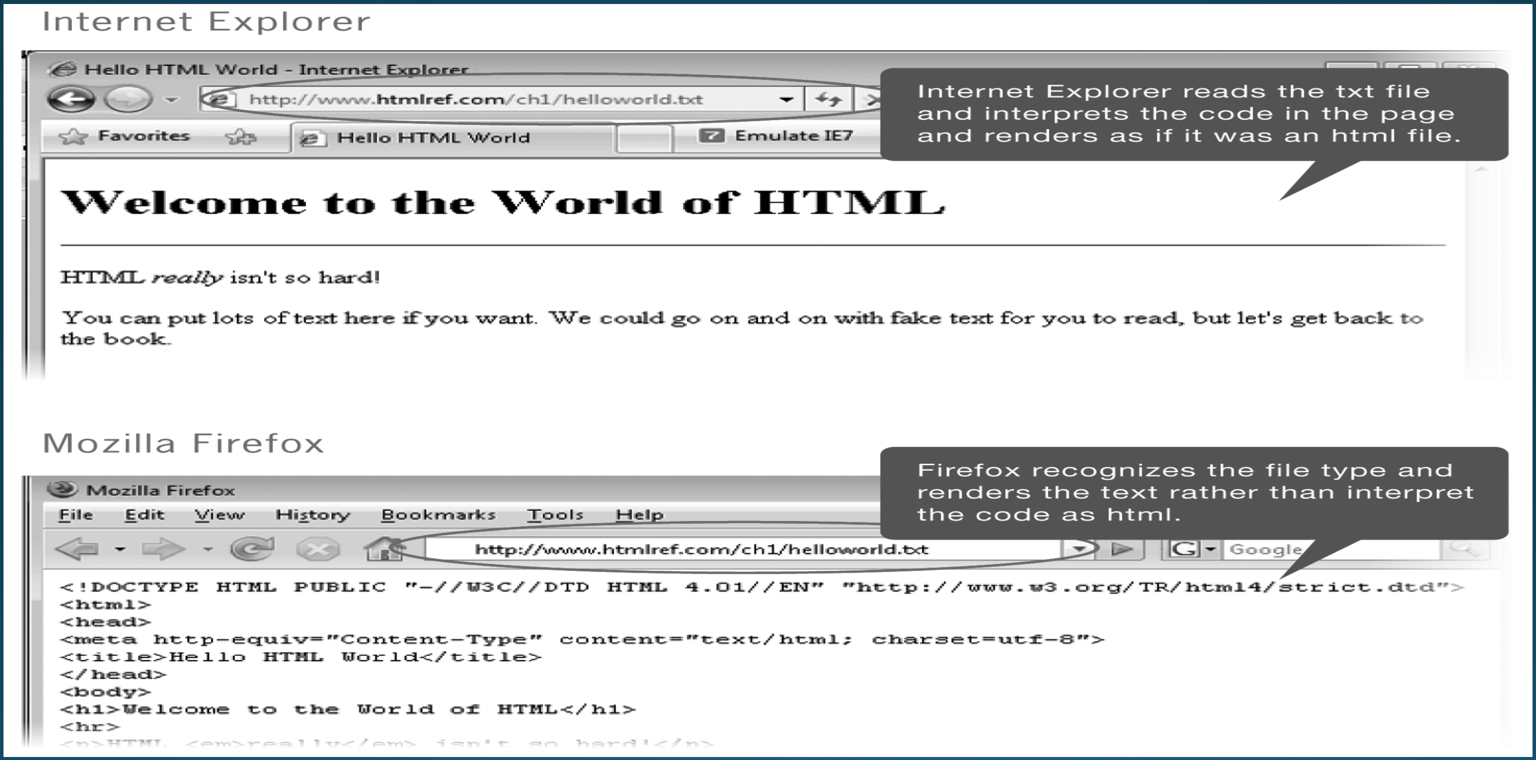
Make sure to serve it with the correct MIME type!

| URL | Status | Domain | Size |
|---|---|---|---|
| GET xhtml5helloworld.xhtml | 200 OK | htmlref.com | 482 B |

Response Headers

```
  Content-Length 482
    Content-Type application/xhtml+xml
Content-Location http://htmlref.com/ch2/xhtml5helloworld.xhtml
   Last-Modified Tue, 27 Oct 2009 02:30:55 GMT
   Accept-Ranges bytes
            Etag "80d95d82ad56ca1:5e8"
```

# It’s not a detail


Internet Explorer

Hello HTML World - Internet Explorer

http://www.htmlref.com/ch1/helloworld.txt

Internet Explorer reads the txt file and interprets the code in the page and renders as if it was an html file.

**Welcome to the World of HTML**

HTML *really* isn't so hard!

You can put lots of text here if you want. We could go on and on with fake text for you to read, but let's get back to the book.

Mozilla Firefox

Firefox recognizes the file type and renders the text rather than interpret the code as html.

http://www.htmlref.com/ch1/helloworld.txt

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Hello HTML World</title>
</head>
<body>
<h1>Welcome to the World of HTML</h1>
<hr>
<p>HTML <em>really</em> isn't so hard!</p>
```

- Sadly people don’t know MIME and trouble can happen when we “fix” things for them - Thomas’ Law of Unintended Consequences

## CSS

- Versions
  - CSS1, CSS-P, CSS2, CSS2.1, CSS3*
  - CSS3 is crazy it is a hodge podge of a million different things of varying states of likelihood of being real or not

## Proper CSS Usage

- Separation of concerns
  - HTML for structure, CSS for style
  - External CSS files (watch downloads)
  - Organization style - alpha, general->specific, etc.
  - Watch out for classitis and repeated IDs
- The dream and reality of the Zen Garden
  - http://www.csszengarden.com/

## Vendor Prefix Fun

- Prefix new features to keep implementations separate until standard
  - -moz , -webkit, -ms, -o
  - Ex: -moz-column-count, -webkit-column-count, column-count
  - Huge redundancy and headache

## !important

- Too many rules, bad tree, inheritance complexity -> things just don’t look right
- !important - Useful but ugly hack that should be done inline or at least last rule for best result
  - Ex: <p style=”!important ruleiwant:valiwant”>
  - Except when it is a !important war

## Suggestions

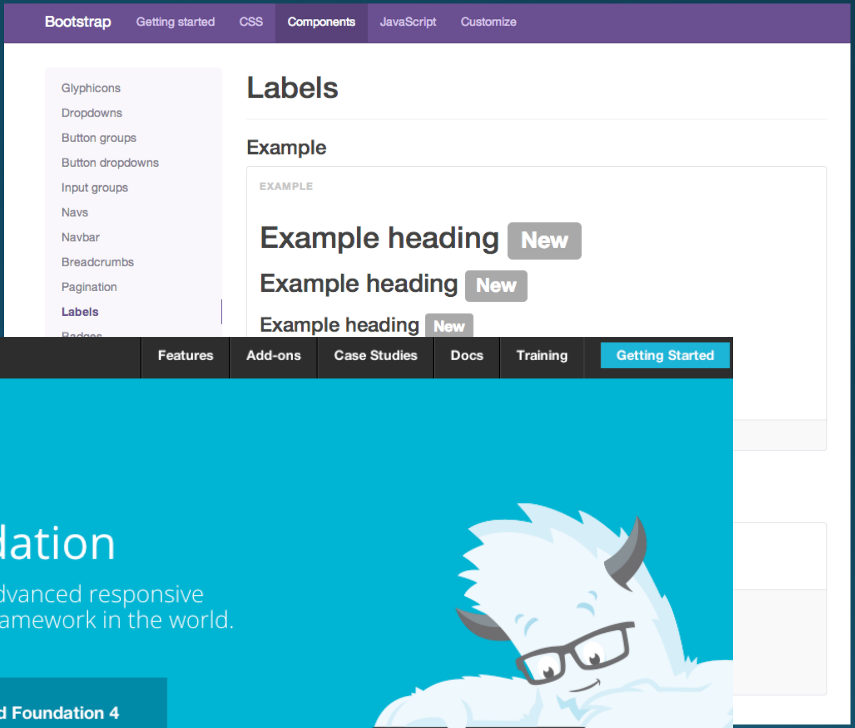
- CSS Validation - http://jigsaw.w3.org/css-validator/
- CSS minification
- CSS bundling
  - Inlining of CSS? Sometimes makes sense

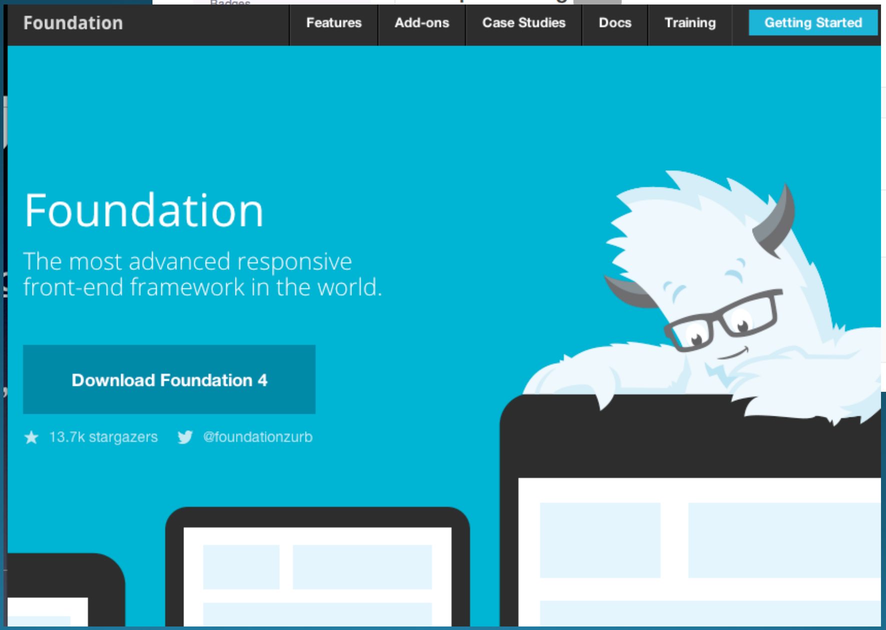
# Suggestion
# Use a Framework

http://getbootstrap.com

http://foundation.zurb.com

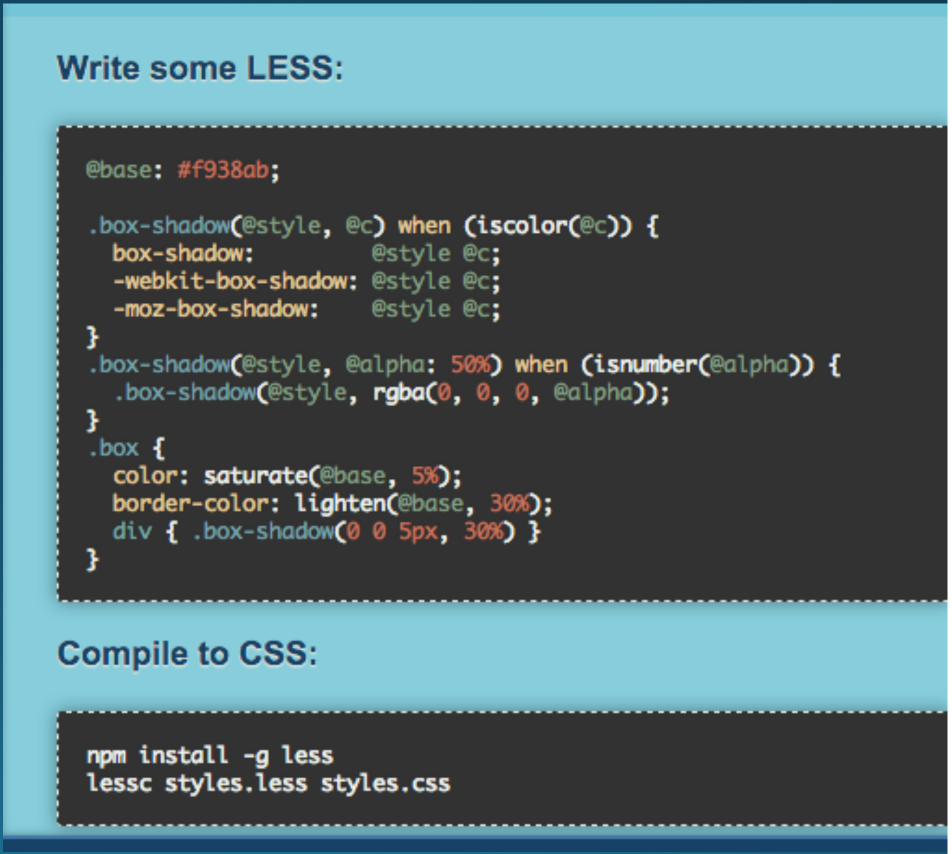
- Concerns
  "Validity", Cruft





## Suggestion - CSS Preprocessors

- Concerns
  - Solving problems that may go away?
  - Solving problems that suggest a deeper problem you have?
- [http://lesscss.org](http://lesscss.org)
- [http://learnboost.github.io/stylus/](http://learnboost.github.io/stylus/)
- [http://sass-lang.com/](http://sass-lang.com/)


Write some LESS:

```
@base: #f938ab;

.box-shadow(@style, @c) when (iscolor(@c)) {
  box-shadow:         @style @c;
  -webkit-box-shadow: @style @c;
  -moz-box-shadow:    @style @c;
}
.box-shadow(@style, @alpha: 50%) when (isnumber(@alpha)) {
  .box-shadow(@style, rgba(0, 0, 0, @alpha));
}
.box {
  color: saturate(@base, 5%);
  border-color: lighten(@base, 30%);
  div { .box-shadow(0 0 5px, 30%) }
}
```

Compile to CSS:

```
npm install -g less
lessc styles.less styles.css
```

## Core JavaScript Discussion Snipped

Presented in a previous lecture

## Suggestion: JS Combining

- Since browser based JS shares same name space combining files doesn’t change anything code wise but improves network delivery because of request reduction
- Production JS files should be combined before deployment
- Combine before minifying in my opinion so you don’t get duplicate names
- Grunt task

## Suggestion: JS Minification

- Reduce source code size for delivery
  - Minor amount of obfuscation
- Common Techniques
  - Remove white space, comments, dead code
  - Rename long names
  - Inline single usage
  - Byte shave (ex. x=x+1 to x++)

## Suggestion: Minification Tools

- Google Closure Compiler https://developers.google.com/closure/compiler/
- UglifyJS http://lisperator.net/uglifyjs/
- Make it part of deployment process
  - GruntJS automation https://github.com/gruntjs/grunt-contrib-uglify

## Minifcation Considerations

- Naming
  - file.js   file.min.js
- Sometimes minifiers break stuff
  - Their bugs, your bad code
  - Sourcemap to allow for in production debug
    - http://www.html5rocks.com/en/tutorials/developertools/sourcemaps/

## Suggestion: Build File Naming

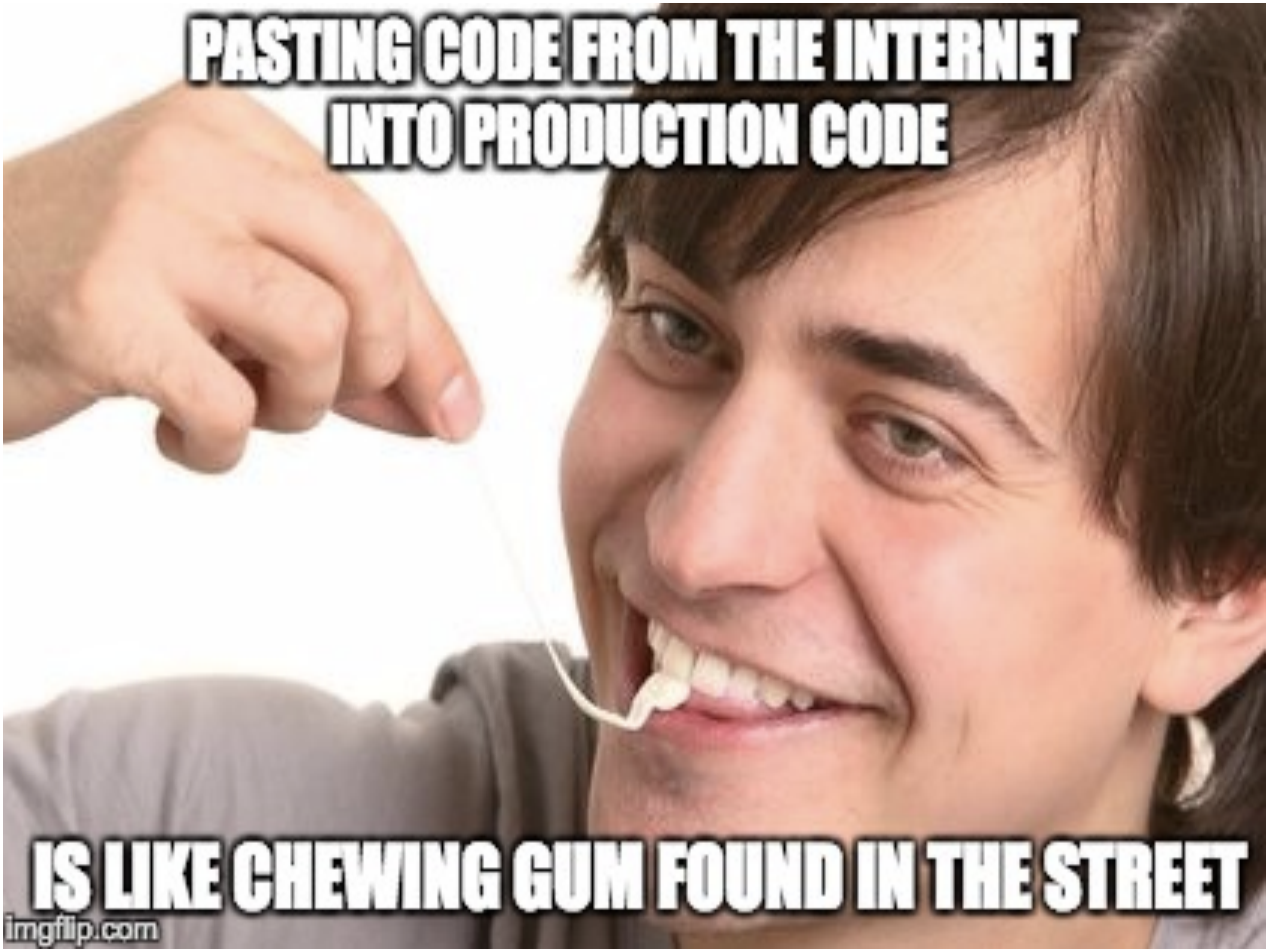
- If you are making somewhat frequent updates we need to be quite careful about browser & proxy caches

```
<script src="myapp.min.js"></script>
<!– is this dangerous? -->

<script
src="myapp.1382403226836.min.js">
</script>
<!-- use timestamp but build id is
that better? -->
```

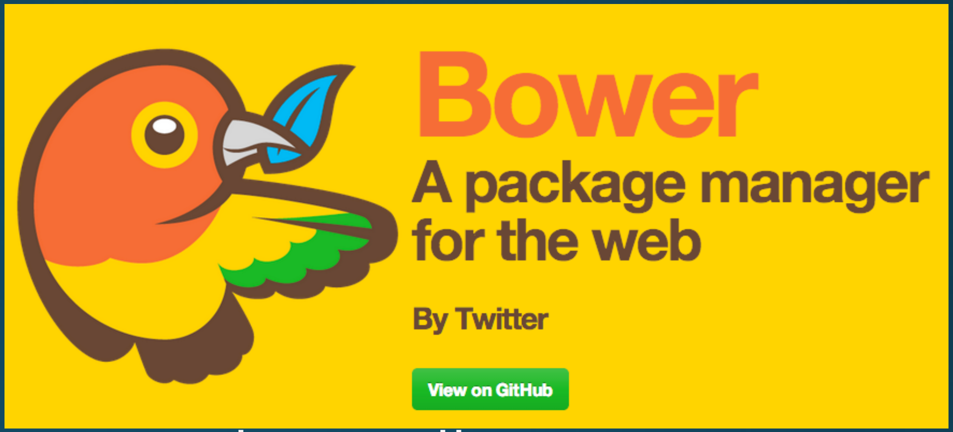
## Suggestion: Package Management

- If you depend on other people's code you have to keep up with it
- Rule of thumb: The more you mix the more volatile things are going to get
- Reactions - include nothing and write everything, upon error exclaim it's not my fault, fork, contribute back


PASTING CODE FROM THE INTERNET INTO PRODUCTION CODE
IS LIKE CHEWING GUM FOUND IN THE STREET
imgflip.com

## Suggestion: Bower

- http://bower.io/
- bower install <x>
- bower install
  consults mainfest in bowser.json and get it all
- Avoid: bower search - do some eval first!



# Suggestion: Scaffolders

- http://yeoman.io/
  - Automates lots of routine tasks
  - Can help speed development - live reloads, staging server, test runni etc.
  - Lots of configuration to learn