

HTML5 Overview

Sang Shin
Founder & Chief Instructor
JPassion.com
“Learn with Passion!”



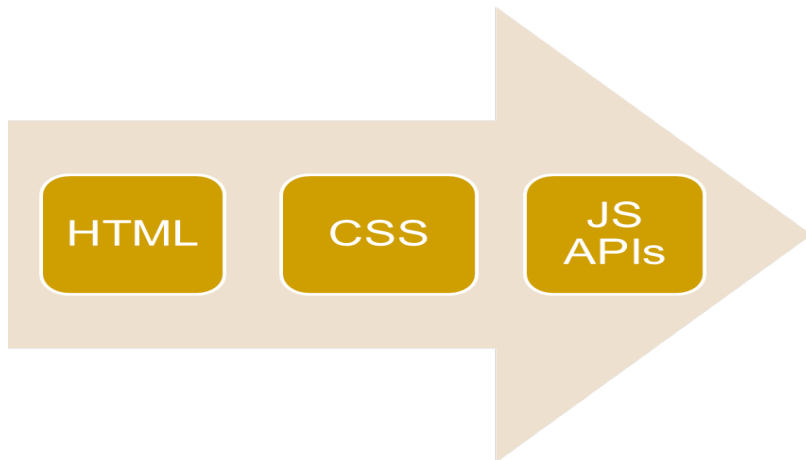
Topics

- What is HTML5?
- Why HTML5?
- HTML5 showcases
- HTML5 features quick overview
- HTML5 support in Browsers
- Building Mobile apps with HTML5
- HTML5-enabled Web Application Architecture
- So how can I get started?

What is HTML5?

What is HTML5?

- Collection of features, technologies, and APIs
- Brings the power of the desktop and the vibrancy of multimedia experience to the web—while amplifying the web's core strengths of interactivity and connectivity
- HTML + CSS3 + JavaScript



How did HTML5 effort started?

- HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).
- WHATWG was working with web forms and applications, and W3C was working with XHTML 2.0.
- In 2006, they decided to cooperate and create a new version of HTML.

Timeline of Web Technologies



Major Design Goals of HTML5

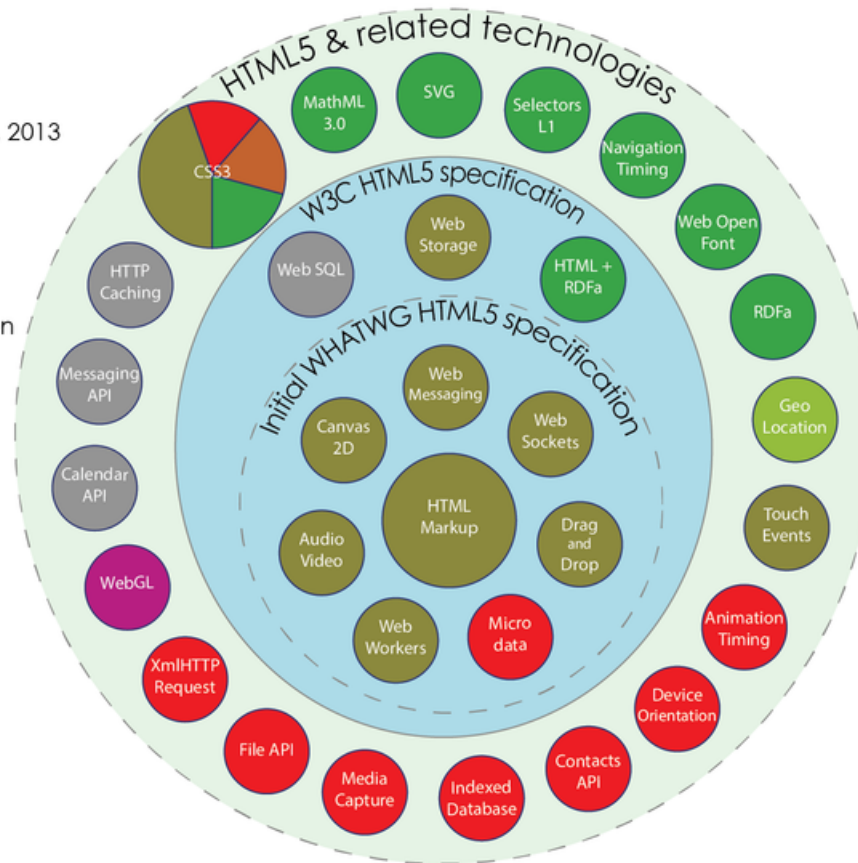
- New features should be based on existing and established standards
 - > HTML, CSS, DOM, and JavaScript
- Eliminate/reduce the need for external plugins (like flash)
- Provide built-in markup replacing custom scripting
- Support wide spectrum of computing devices
- Mobile friendly

HTML5 is a set of many technologies

HTML5

Taxonomy & Status on January 20, 2013

- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated



by Sergey Mavrody (CC) BY · SA

Why HTML5?

A few facts on Browser

- Browser is becoming the application platform of choice
 - > “browser apps” over “desktop apps”: Google apps, Gmail, Games
 - > “browser apps” over “native mobile apps” (Mobile)
- Innovations on browser has been huge
 - > The performance of JavaScript engine increased hundred-fold
- Browser only platforms
 - > Chrome OS, Firefox OS
 - > Fast, secure, reliable, cloud-enabled
 - > No overhead of general-purpose OS platform
- Browser-only devices
 - > ChromeBook

Why HTML5?

- Offline and Storage
 - > Aside from letting apps **retain their states and hold data without a server**, offline features have the bonus of **improving an app's performance** by storing data in the cache or making data persistent between user sessions and page reloading
- Better user experience
 - > HTML5 enables web apps to be **more responsive**, creating a user experience that **rivals that of their desktop counterparts**
- Easier development
 - > HTML5 simplifies your development cycle by letting you use the **same technology stack across multiple platforms and devices**
 - > HTML5 technologies—JavaScript, CSS, and HTML—are easier to learn

Why HTML5?

- Broad reach
 - > HTML5 is broadly supported on all modern desktop browsers and major mobile devices. No other technology can offer the same ubiquity
- Security
 - > Native support in browsers for multimedia and other capabilities reduces the need for plugins, some of which have introduced vulnerabilities to the browser
- Lower cost and easier maintenance
 - > HTML5-enabled web applications are cheaper to run across multiple platforms, they're easier to maintain, and they make specific devices irrelevant. All these features not only lower costs, but also improve productivity.

HTML5 Adoption: Everyone is on it!

- “The world is moving to HTML5”
 - > Steve Jobs (Apple)
- "The world is pushing down HTML 5 path and so are we."
 - > Steve Ballmer (Microsoft)
- “The web has not seen this level of transformation, this level of acceleration, in the past 10 years... we're betting big on HTML5”
 - > Vic Gundotra (Google)

No more plugins are required

- http://readwrite.com/2011/11/08/steve_jobs_wins_adobe_to_give_up_mobile_flash_for

The screenshot shows a web browser window with multiple tabs. The active tab is titled "Steve Jobs Wins: / x". The address bar shows the URL "readwrite.com/2011/11/08/steve_jobs_wins_adobe_to_give_up_mobile_flash_for". The page content includes a navigation bar with "Sections", "Search", "Follow us", and "Share". The main article is titled "Steve Jobs Wins: Adobe to Give Up Mobile Flash for HTML5" by Marshall Kirkpatrick, dated November 8th, 2011. The article features a red square icon with a white Adobe logo. Below the icon is the alt text "Alt title: A Win for the Web." and a paragraph of text: "Sources close to the news have told ZDNet reporter Jason Perlow tonight that Adobe will announce soon that it has given up on the development of mobile flash and will increase its investment in supporting HTML5. The company will say, according to an email". To the left of the main article is a sidebar with "All Articles" and a list of recent articles with timestamps. To the right is a "Sites We Love" section with a link to "Technology on HuffingtonPost.com" and a video player.

Steve Jobs Wins: / x

The Web Can Do x Google Maps x Chrome Web Stor x WebGL Electric Fl x

readwrite.com/2011/11/08/steve_jobs_wins_adobe_to_give_up_mobile_flash_for

JPassion.com - ... My Bookmarks Google Other bookmarks

Sections Search Follow us Share

All Articles

Wed, 13 Feb 2013

PLAY · 6:05 PM
Slacker Redesign Sets Streaming-Music Battle With Spotify & Pandora

· 2:04 PM
Intel Confirms Its TV Plans - There's No Way It's Going To Work

· 11:31 AM
MacBook Pro Retina Models Get Price Cut & Faster Chips


MOBILE · 11:15 AM
HP To Adopt Android For Upcoming Mobile Devices

MOBILE · 8:11 AM
Draeger Maker Opens

Steve Jobs Wins: Adobe to Give Up Mobile Flash for HTML5

Marshall Kirkpatrick · November 8th, 2011

+1 63 Tweet 1,217 Share 324 Like 0



Alt title: A Win for the Web.

Sources close to the news have told ZDNet reporter Jason Perlow tonight that Adobe will announce soon that it has given up on the development of mobile flash and will increase its investment in supporting HTML5. The company will say, according to an email

119 HP To Adopt Android For Upcoming Mobile Devices

11 Exploratorium's Experience Experts Deliver Awesome iPad App

9 Why Oracle Fusion Doesn't Excite Customers

Sites We Love

Technology on HuffingtonPost.com

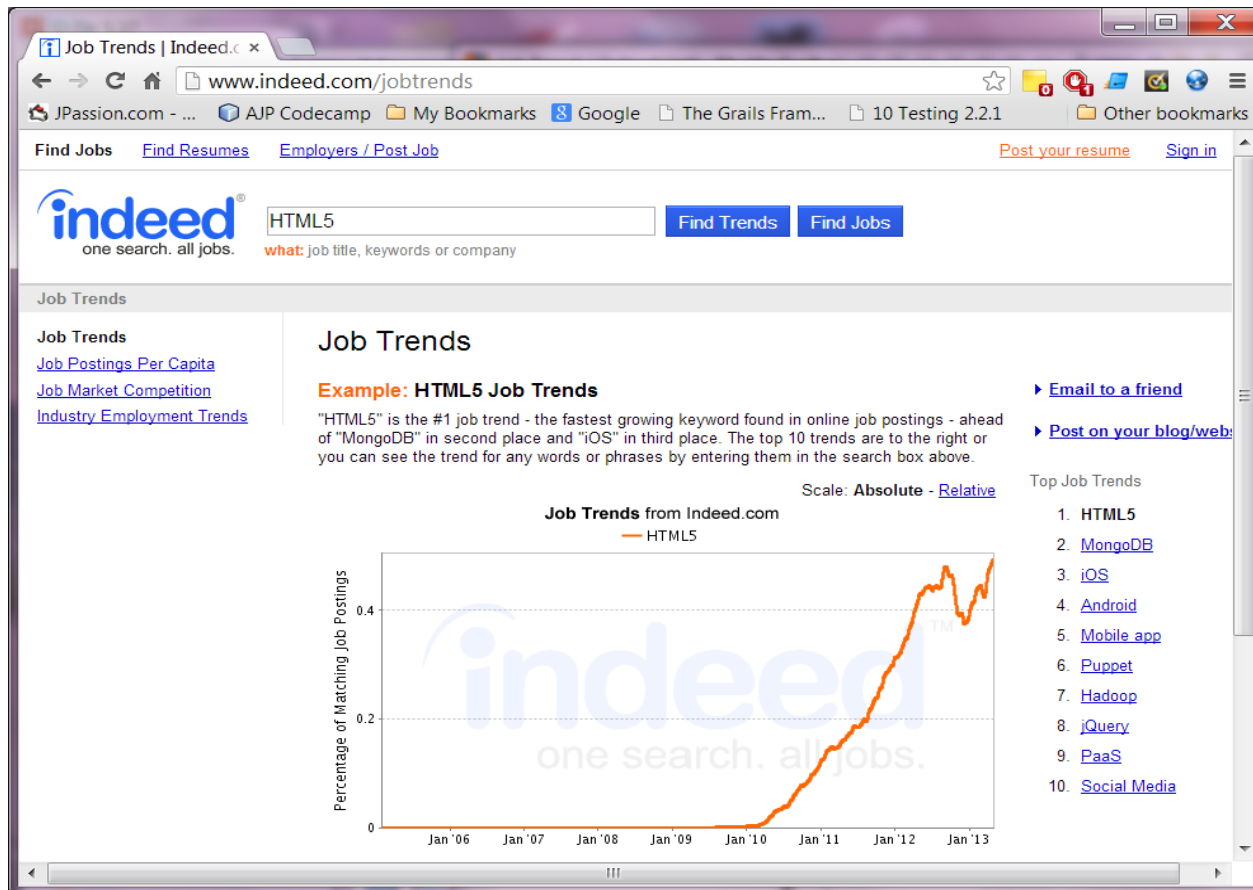
VIDEO: Think YOU Filmed Some Good Stuff On Your GoPro?
36 MINUTES AGO

This Is What The Near-Miss Asteroid Would See As It Grazes Earth
AN HOUR AGO

Xbox Co-Founder Lets Loose At Microsoft's Five Years Of Failure
2 HOURS AGO

HTML5 is No#1 in Job Trends

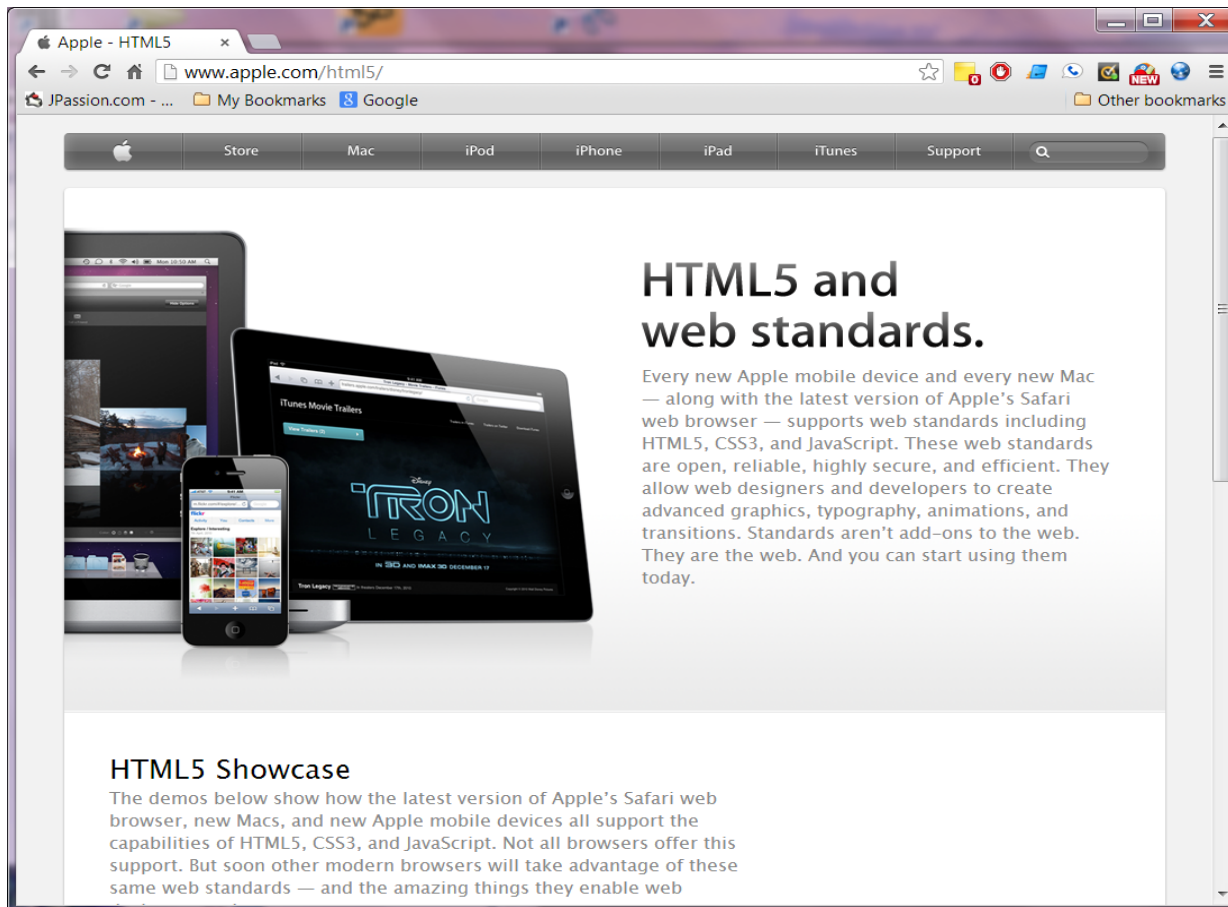
- <http://www.indeed.com/jobtrends>



HTML5 Showcases

Apple HTML5 Showcases

- <http://www.apple.com/html5/> (no longer available)



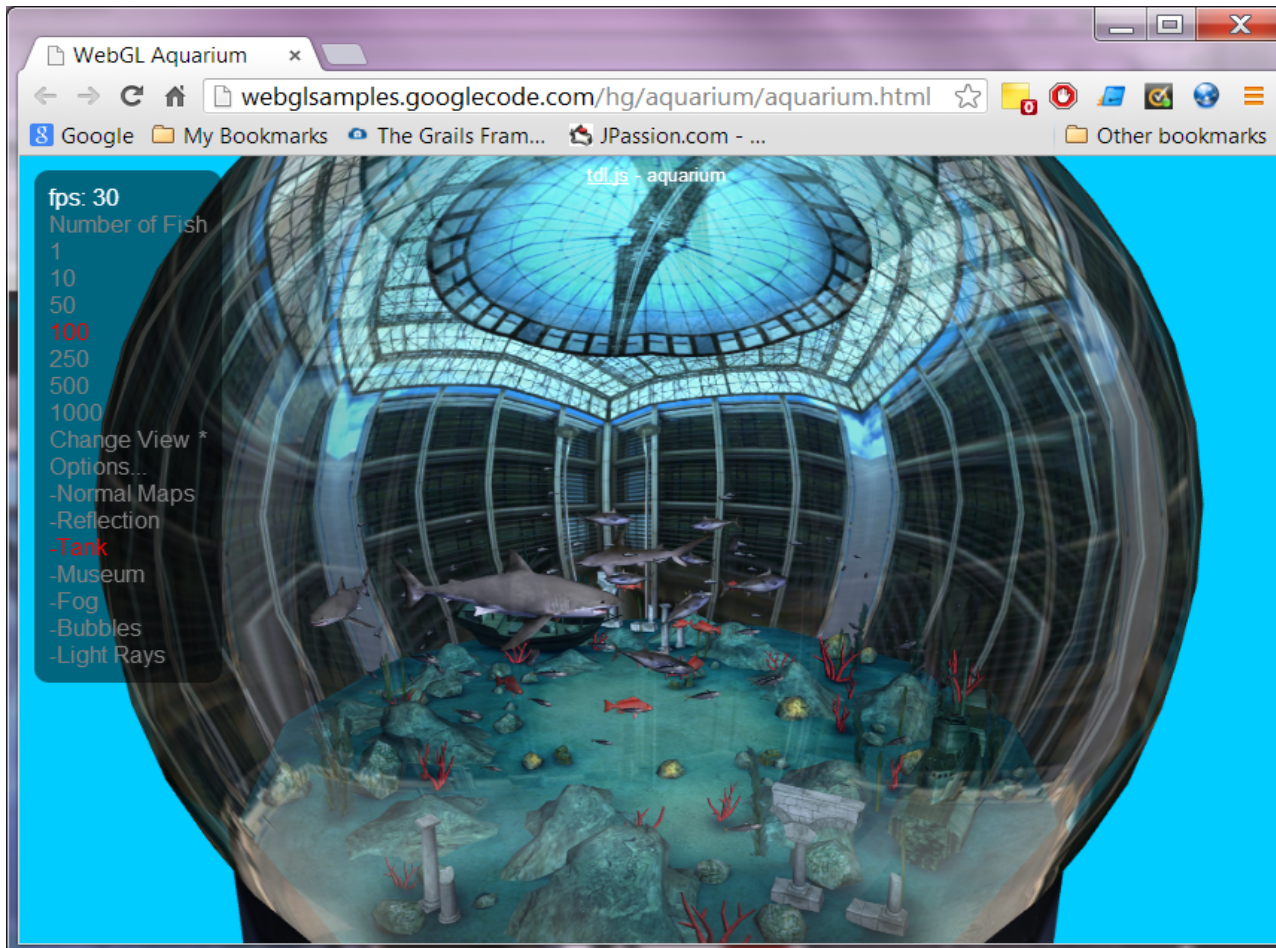
Everything Google is HTML5 based

- <http://chrome.angrybirds.com/>



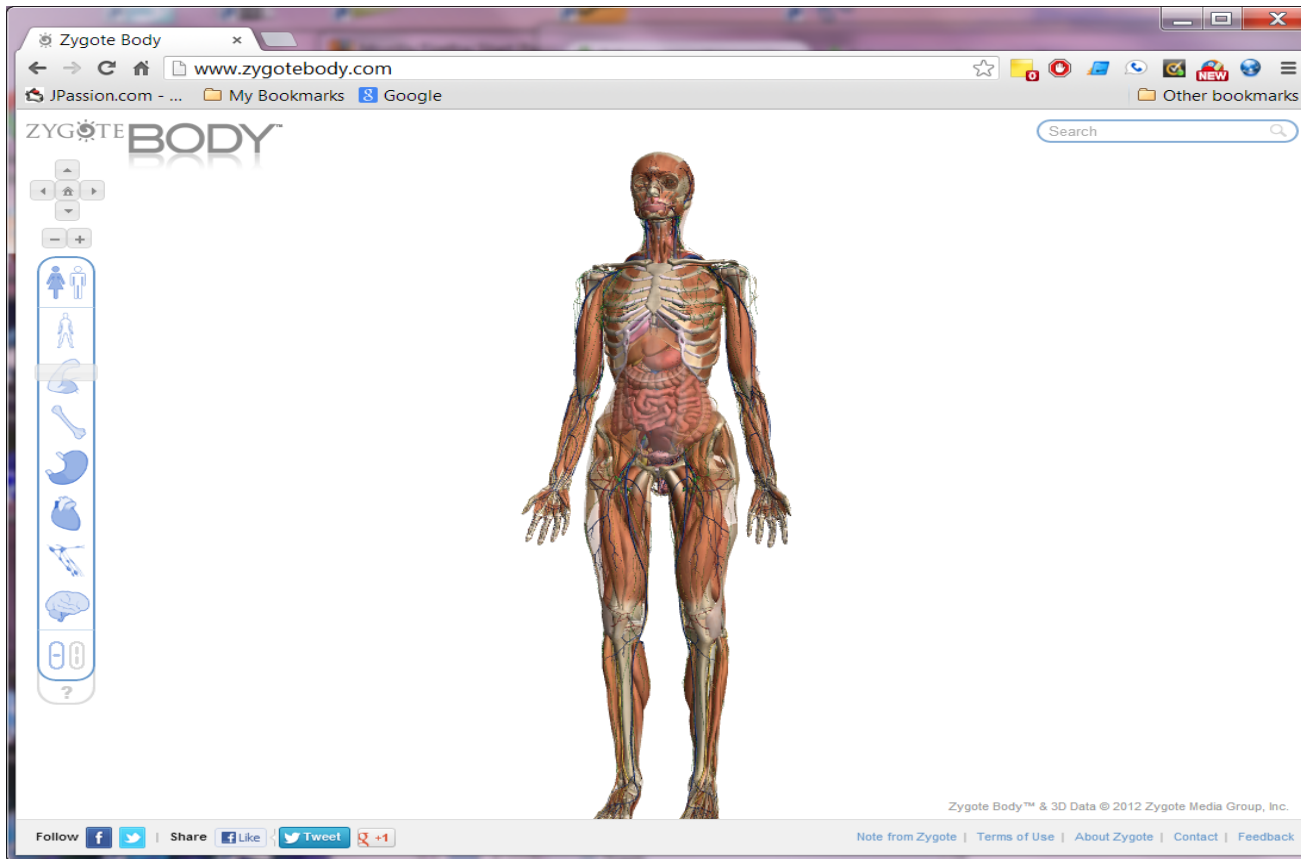
WebGL

- <http://webglsamples.googlecode.com/hg/aquarium/aquarium.html>



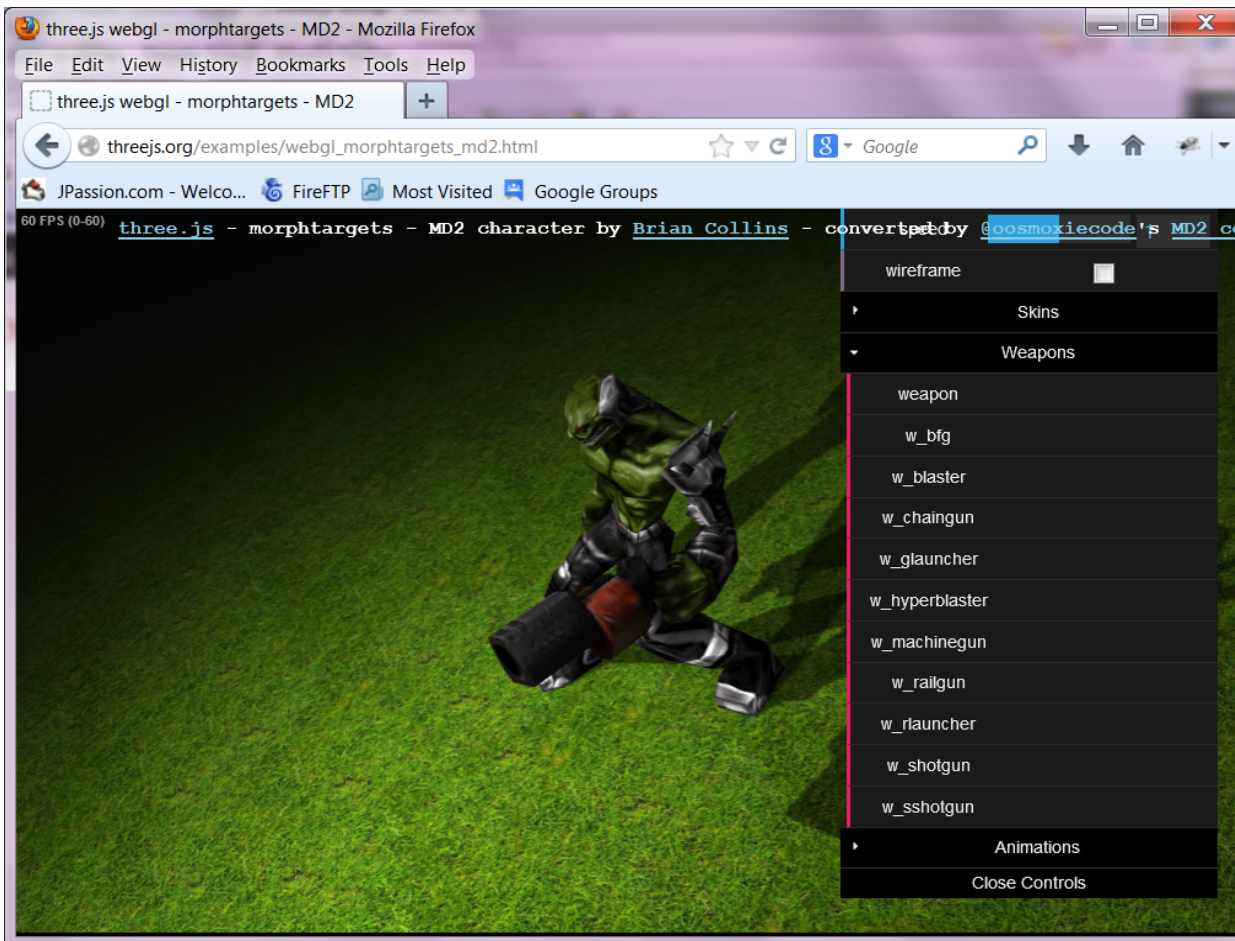
Google Body

- <http://www.zygotebody.com/>



Three.js (WebGL Library)

- http://threejs.org/examples/webgl_morphtargets_md2.html



HTML5 Features Quick Overview

Semantics

- New media elements.
- New structural elements.
- New semantics for internationalization.
- New link relation types.
- New attributes.
- New form types.
- New microdata syntax for additional semantics.



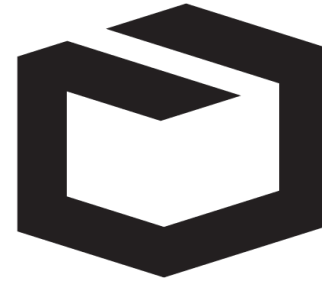
Presentation

- Styling with CSS 3
 - > 2D Transformations & Transitions
 - > 3D Transforms
 - > WebFonts



Graphics

- Canvas element
- Web GL
- SVG



Connectivity

- Web Socket
 - > Full-duplex bidirectional communication channel
- Cross-domain messaging
- Server-side events
- XMLHttpRequest2



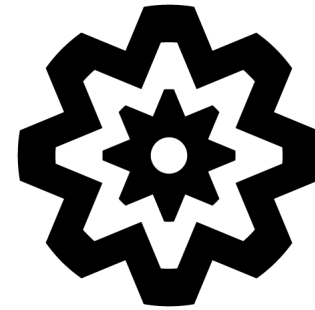
Audio/Video

- You can access, control and manipulate timeline data and network states of the audio/video files



Performance

- Web apps are now able to rival performance of native and desktop applications
- Web Workers
 - > Background processing



Storage



- Application cache
- Local storage
- Web SQL (Deprecated)
- IndexedDB

File Access

- Provides very powerful APIs to interact with binary data and a user's local file system



HTML5 Support in Browsers

Check Browser Support from <http://caniuse.com>

The screenshot shows the CanIUse.com website with a search for 'websocket'. The search results show 1 result found. The main content area displays a compatibility table for Web Sockets, which is marked as a 'Working Draft'. The table includes columns for various browsers and their versions, with support status indicated by colored cells (green for supported, red for not supported, yellow for partially supported, and grey for support unknown). A legend explains the color coding. Usage statistics are also provided, showing that 57.1% of users have support, 4.64% have partial support, and a total of 61.74% have some level of support. A notes section at the bottom explains that partial support refers to older versions of the protocol or disabled implementations.

Can I use... Suggestions Feed Twitter 487
Compatibility tables for support of HTML5, CSS3, SVG and more in desktop and mobile browsers. Flattr

Latest update: *Three new features added: Strict Transport Security, CSS outline property and download attribute (December 3, 2012)*

Search: websocket x
1 result found

Index Tables Import stats FAQ Resources Embed

Compatibility tables Browser comparison

Show options Supported Not supported Partially supported Support unknown

Web Sockets - **Working Draft**
Bidirectional communication technology for web apps

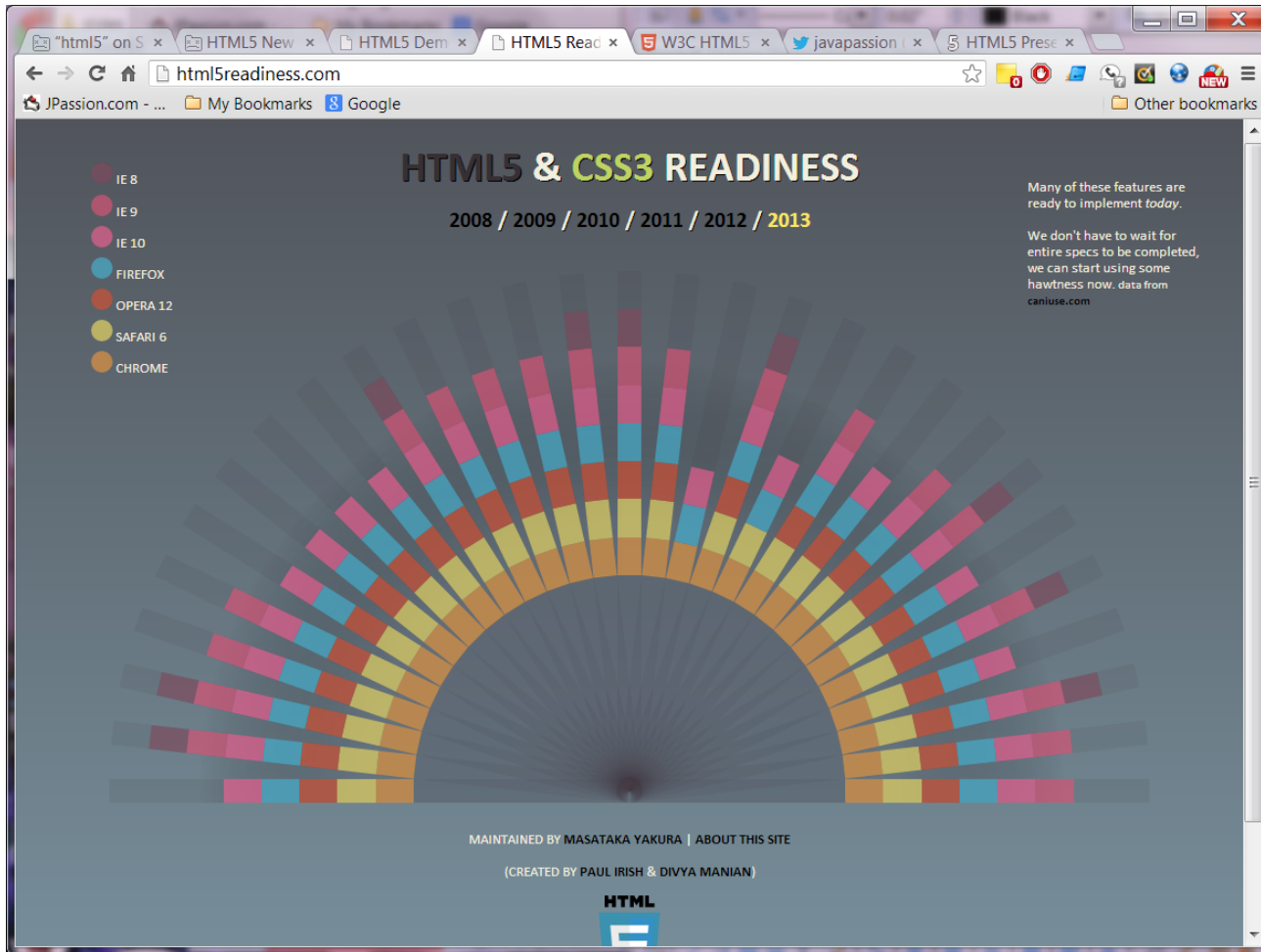
*Usage stats: Global
Support: 57.1%
Partial support: 4.64%
Total: 61.74%

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	BlackBerry Browser
								2.1	
								2.2	
						3.2		2.3	
	7.0					4.0-4.1		3.0	
	8.0	16.0				4.2-4.3		4.0	
	9.0	17.0	23.0	5.1		5.0-5.1		4.1	
Current	10.0	18.0	24.0	6.0	12.1	6.0	5.0-7.0	4.2	7.0
Near future		19.0	25.0		12.5				10.0
Farther future		20.0	26.0						

Notes Known issues (0) Resources (4) Feedback Edit on GitHub

Partial support refers to the websockets implementation using an older version of the protocol and/or the implementation being disabled by default (due to security issues with the older protocol).

Check Browser Support from <http://html5readiness.com/>



Your Browser's HTML5 Support Score from http://HTML5test.com

The screenshot shows the HTML5test.com website. The main heading is "THE HTML5 TEST - HOW WELL DOES YOUR BROWSER SUPPORT HTML5?". Below this, there are navigation tabs: "your browser", "other browsers", "compare", "news", and "about the test". The "your browser" tab is selected.

The main content area displays "your browser scores" with a large "448" and "AND 13 BONUS POINTS" below it, followed by "out of a total of 500 points". To the right of the score, there are social media sharing options: 8,898 tweets, 13k likes, and 4.4k shares.

Below the score, it says "You are using Chrome 24 on Windows 7" with a "Correct?" checkbox and a checkmark. A section titled "Parsing rules" shows "+2 bonus points" and a score of "10". The rules listed are:

Rule	Result
<!DOCTYPE html> triggers standards mode	Yes ✓
HTML5 tokenizer	Yes ✓
HTML5 tree building	Yes ✓

Below the table, there is a note: "HTML5 defines rules for embedding SVG and MathML inside a regular HTML document. Support for SVG and MathML is not required though, so bonus points are awarded if your browser supports embedding these two technologies."

On the right side, there is an "HTML5" logo and a paragraph explaining the test: "The HTML5 test score is an indication of how well your browser supports the upcoming HTML5 standard and related specifications. Even though the specification isn't finalized yet, all major browser manufacturers are making sure their browser is ready for the future. Find out which parts of HTML5 are already supported by your browser today and compare the results with other browsers."

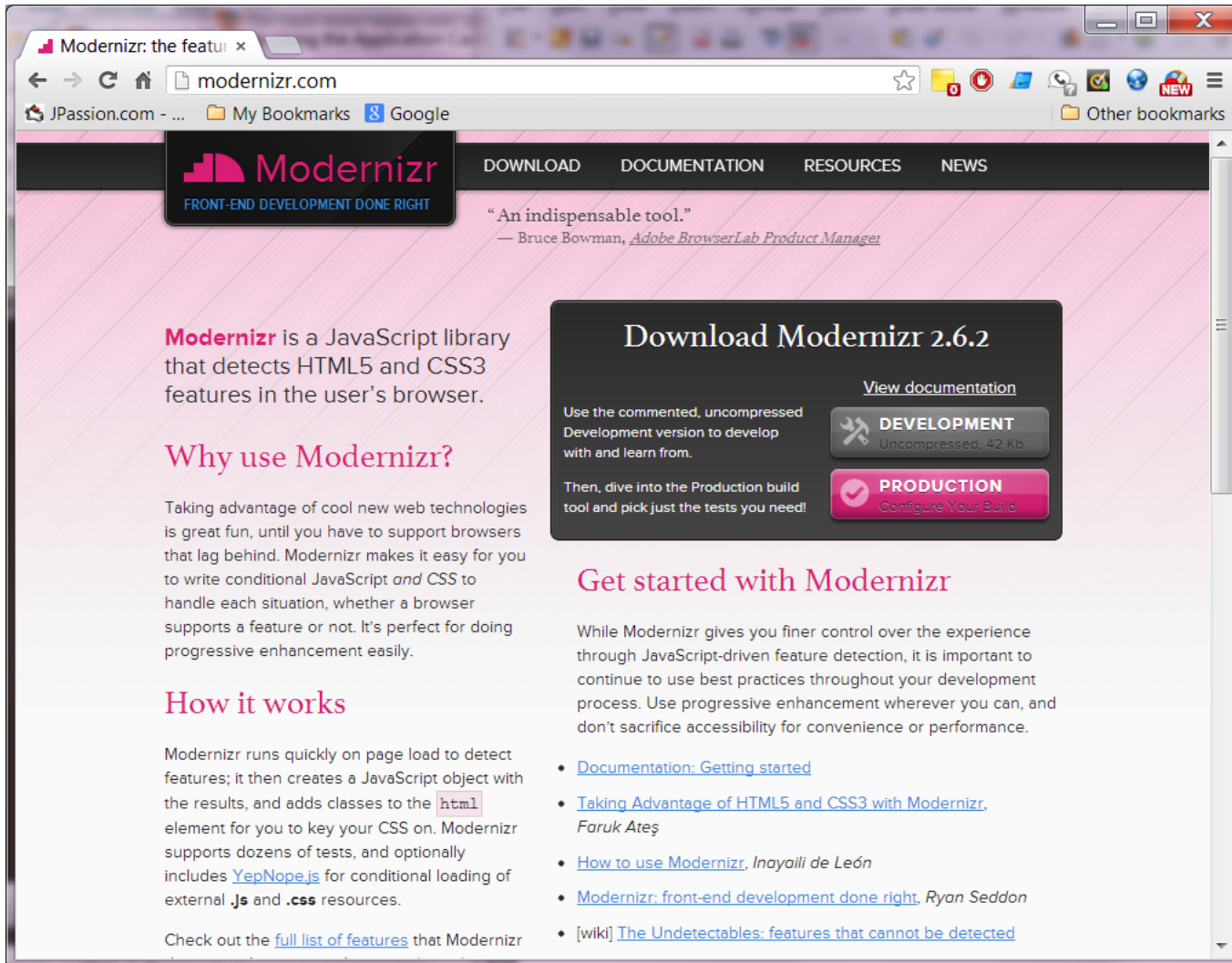
At the bottom right, another paragraph states: "The HTML5 test does not try to test all of the new features offered by HTML5, nor does it try to test the functionality of each feature it does detect. Despite these shortcomings we hope that by quantifying the level of support users and web developers will get an idea of how hard the browser manufacturers work on improving their browsers and the web as a development"

Browser Compatibility Score from <http://html5test.com/compare/browser/chrome24/ff18/ie10.html>

The screenshot shows the HTML5 Test website interface. The browser's address bar displays the URL `http://html5test.com/compare/browser/chrome24/ff18/ie10.html`. The page title is "THE HTML5 TEST - HOW WELL DOES YOUR BROWSER SUPPORT HTML5?". Navigation tabs include "your browser", "other browsers", "compare", "news", and "about the test". The "compare" tab is active, and sub-tabs "browsers" and "features" are visible. The "browsers" sub-tab is selected, showing a comparison of three browsers: Chrome 24 (score 448, 13 bonus points), Firefox 18 (score 393, 10 bonus points), and Internet Explorer 10 (score 320, 6 bonus points). Below this, two sections are shown: "Parsing rules" and "Canvas". Each section has a table with columns for the three browsers, showing "Yes" with a green checkmark for all features listed.

	Chrome 24	Firefox 18	Internet Explorer 10
browsers	448 13 bonus points	393 10 bonus points	320 6 bonus points
Parsing rules	10	10	10
<!DOCTYPE html> triggers standards mode »	Yes ✓	Yes ✓	Yes ✓
HTML5 tokenizer »	Yes ✓	Yes ✓	Yes ✓
HTML5 tree building »	Yes ✓	Yes ✓	Yes ✓
SVG in text/html »	Yes ✓	Yes ✓	Yes ✓
MathML in text/html »	Yes ✓	Yes ✓	Yes ✓
Canvas	20	20	20
canvas element »	Yes ✓	Yes ✓	Yes ✓
2D context »	Yes ✓	Yes ✓	Yes ✓
Text »	Yes ✓	Yes ✓	Yes ✓

Modernizr from <http://modernizr.com/>



The screenshot shows a browser window with the URL modernizr.com. The page features a dark navigation bar with links for [DOWNLOAD](#), [DOCUMENTATION](#), [RESOURCES](#), and [NEWS](#). The main content area has a pink background with diagonal lines. On the left, there is a section titled "Modernizr" with the tagline "FRONT-END DEVELOPMENT DONE RIGHT". Below this, a quote reads: "An indispensable tool." — Bruce Bowman, *Adobe BrowserLab Product Manager*. The main text describes Modernizr as a JavaScript library that detects HTML5 and CSS3 features. A central dark box highlights the "Download Modernizr 2.6.2" section, offering "DEVELOPMENT" (Uncompressed, 42 Kb) and "PRODUCTION" (Configure Your Build) options. The right side of the page includes a "Get started with Modernizr" section with a list of links to documentation and articles.

Modernizr: the featur x

modernizr.com

JPassion.com - ... My Bookmarks Google Other bookmarks

Modernizr
FRONT-END DEVELOPMENT DONE RIGHT

[DOWNLOAD](#) [DOCUMENTATION](#) [RESOURCES](#) [NEWS](#)

"An indispensable tool."
— Bruce Bowman, *Adobe BrowserLab Product Manager*

Modernizr is a JavaScript library that detects HTML5 and CSS3 features in the user's browser.

Why use Modernizr?

Taking advantage of cool new web technologies is great fun, until you have to support browsers that lag behind. Modernizr makes it easy for you to write conditional JavaScript and CSS to handle each situation, whether a browser supports a feature or not. It's perfect for doing progressive enhancement easily.

How it works

Modernizr runs quickly on page load to detect features; it then creates a JavaScript object with the results, and adds classes to the `html` element for you to key your CSS on. Modernizr supports dozens of tests, and optionally includes [YepNope.js](#) for conditional loading of external `.js` and `.css` resources.

Check out the [full list of features](#) that Modernizr

Download Modernizr 2.6.2

[View documentation](#)

Use the commented, uncompressed Development version to develop with and learn from.

Then, dive into the Production build tool and pick just the tests you need!

DEVELOPMENT
Uncompressed, 42 Kb

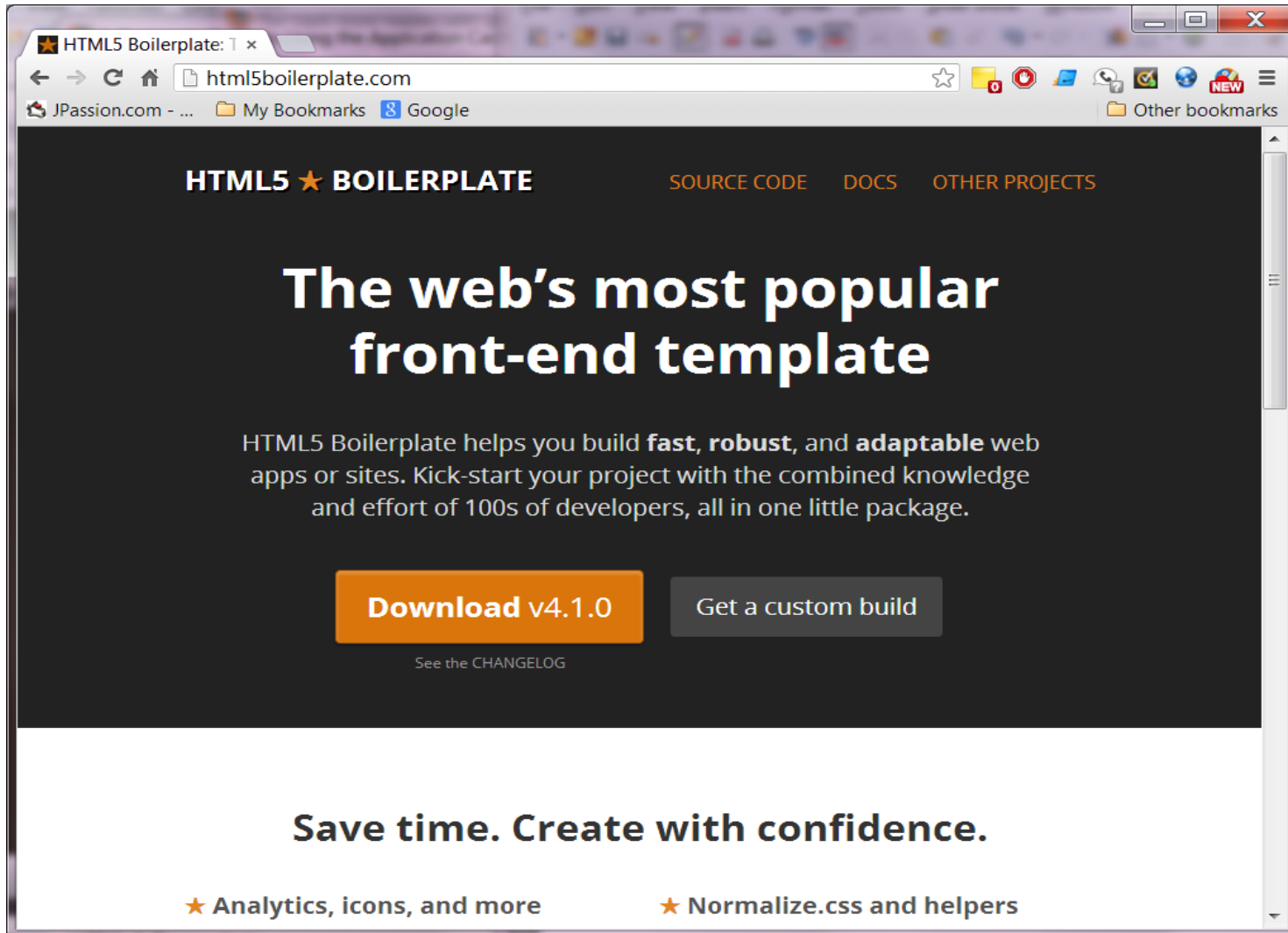
PRODUCTION
Configure Your Build

Get started with Modernizr

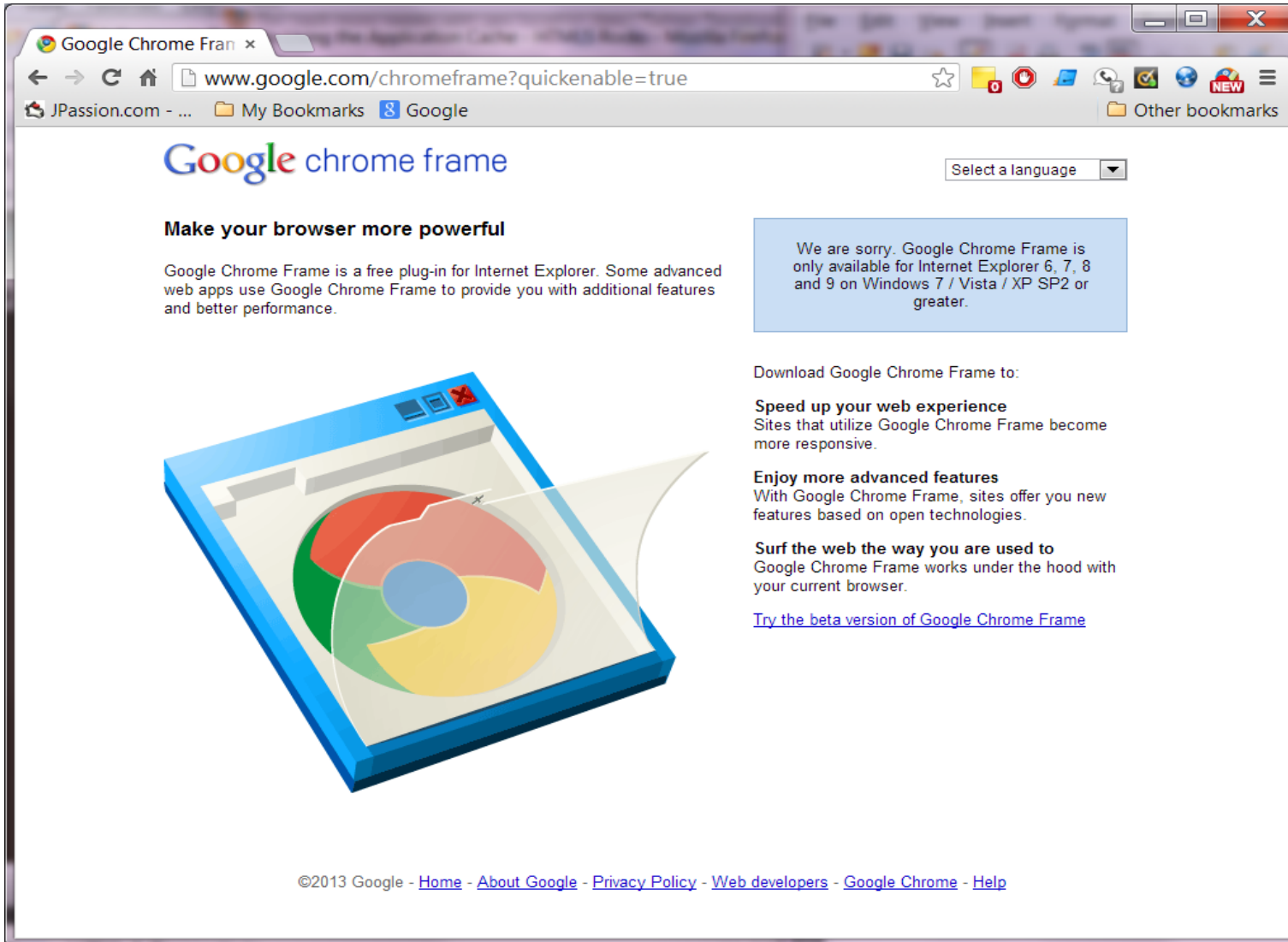
While Modernizr gives you finer control over the experience through JavaScript-driven feature detection, it is important to continue to use best practices throughout your development process. Use progressive enhancement wherever you can, and don't sacrifice accessibility for convenience or performance.

- [Documentation: Getting started](#)
- [Taking Advantage of HTML5 and CSS3 with Modernizr](#), Faruk Ateş
- [How to use Modernizr](#), Inayaili de León
- [Modernizr: front-end development done right](#), Ryan Seddon
- [wiki] [The Undetectables: features that cannot be detected](#)

HTML5 Boilerplate from <http://html5boilerplate.com/>



Chrome Frame from <http://google.com/chromeframe>



The screenshot shows a browser window with the address bar containing www.google.com/chromeframe?quickenable=true. The page title is "Google Chrome Frame". The main content includes the Google logo, a language selection dropdown, and several sections of text and images. A blue box on the right contains a message about browser compatibility. At the bottom, there is a copyright notice and a list of links.

Google chrome frame Select a language

Make your browser more powerful

Google Chrome Frame is a free plug-in for Internet Explorer. Some advanced web apps use Google Chrome Frame to provide you with additional features and better performance.

We are sorry. Google Chrome Frame is only available for Internet Explorer 6, 7, 8 and 9 on Windows 7 / Vista / XP SP2 or greater.

Download Google Chrome Frame to:

Speed up your web experience
Sites that utilize Google Chrome Frame become more responsive.

Enjoy more advanced features
With Google Chrome Frame, sites offer you new features based on open technologies.

Surf the web the way you are used to
Google Chrome Frame works under the hood with your current browser.

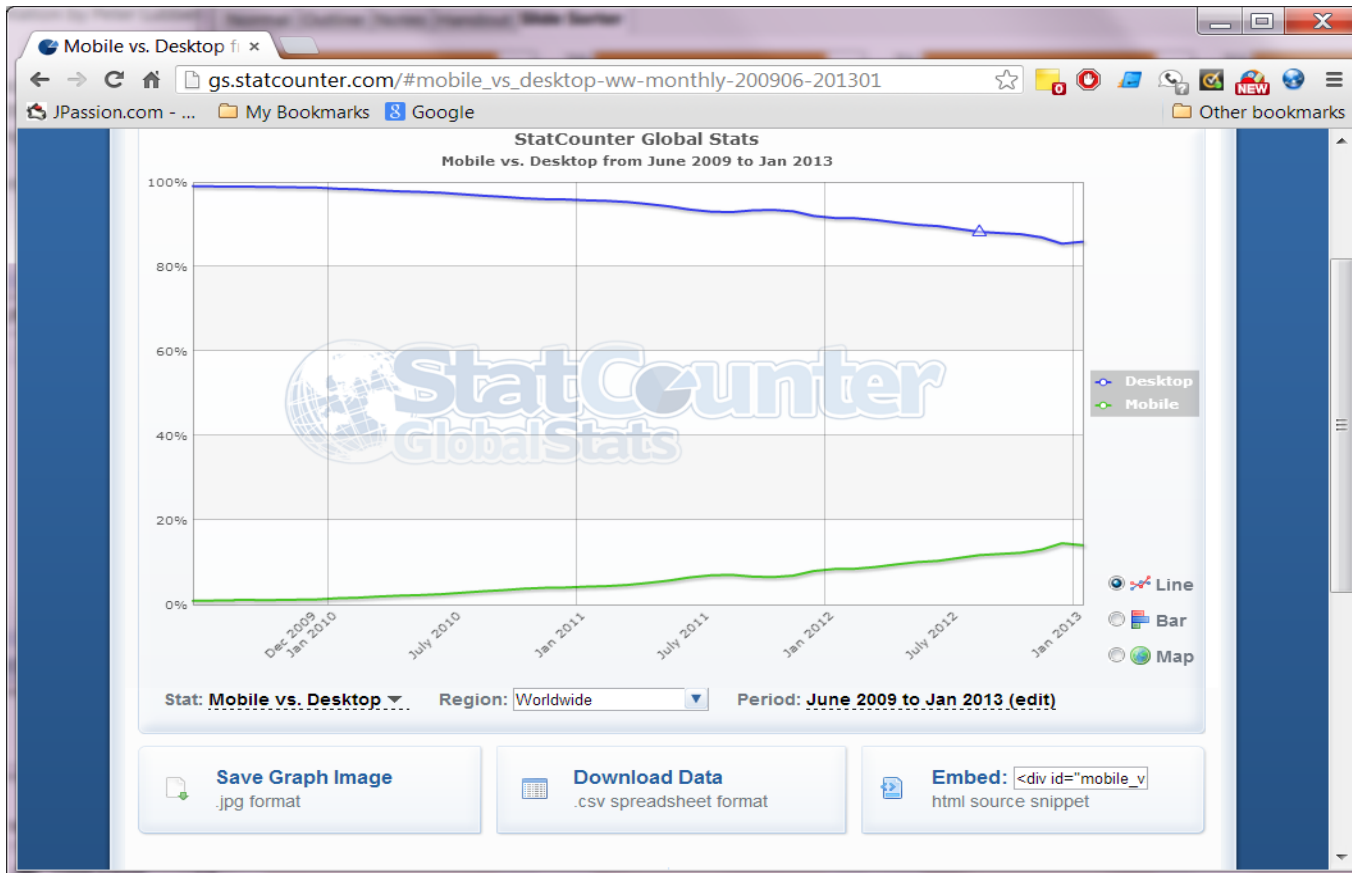
[Try the beta version of Google Chrome Frame](#)

©2013 Google - [Home](#) - [About Google](#) - [Privacy Policy](#) - [Web developers](#) - [Google Chrome](#) - [Help](#)

HTML5 and Mobile Apps

Desktop Browser vs Mobile Browser

- http://gs.statcounter.com/#mobile_vs_desktop-ww-monthly-200906-201301



HTML5 Support on Mobile Platforms

The screenshot shows the 'The HTML5 Test' website. The browser's address bar displays 'html5test.com/compare/browser/android40/ios60/wp80.html'. The page title is 'THE HTML5 TEST - HOW WELL DOES YOUR BROWSER SUPPORT HTML5?'. Navigation tabs include 'your browser', 'other browsers', 'compare', 'news', and 'about the test'. The 'compare' tab is active, showing a comparison of HTML5 support across three mobile platforms: Android 4.0, iOS 6.0, and Windows Phone 8. The 'browsers' section is highlighted, and the 'features' sub-tab is selected. The comparison table shows scores for various HTML5 features, with 'Parsing rules' and 'Canvas' being the most detailed sections shown.

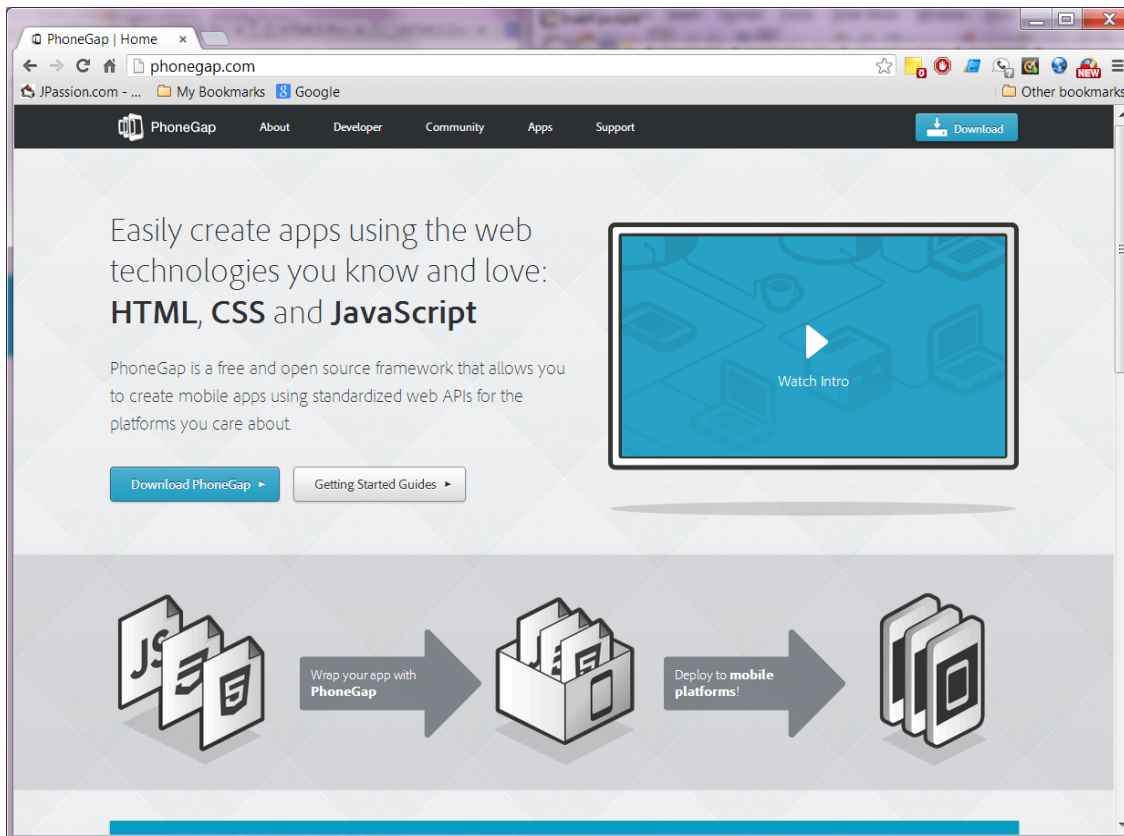
	Android 4.0	iOS 6.0	Windows Phone 8
browsers	297 3 bonus points	386 9 bonus points	320 6 bonus points
Parsing rules	10	10	10
<!DOCTYPE html> triggers standards mode »	Yes ✓	Yes ✓	Yes ✓
HTML5 tokenizer »	Yes ✓	Yes ✓	Yes ✓
HTML5 tree building »	Yes ✓	Yes ✓	Yes ✓
SVG in text/html »	Yes ✓	Yes ✓	Yes ✓
MathML in text/html »	Yes ✓	Yes ✓	Yes ✓
Canvas	20	20	20
canvas element »	Yes ✓	Yes ✓	Yes ✓
2D context »	Yes ✓	Yes ✓	Yes ✓
Text »	Yes ✓	Yes ✓	Yes ✓
Video	21	21	30

Why HTML5 Mobile Apps (vs. Native Mobile Apps)?

- Easier to access and use for your users
 - > No need to download Apps (unlike native Apps)
- Cross-platform
 - > A single codebase for all devices (unlike native Apps)
 - > No need to write device-specific code
 - > Easier to maintain
- Standard-based
 - > Broad adoption
 - > Larger Ecosystem
 - > Safer path for the future
- Web Search'able and bookmark'able

PhoneGap bridges the gap

- Allows you to create mobile apps using HTML5 for the platforms you care about



HTML5-enabled Web Application Architecture

Web App Architecture Requirements

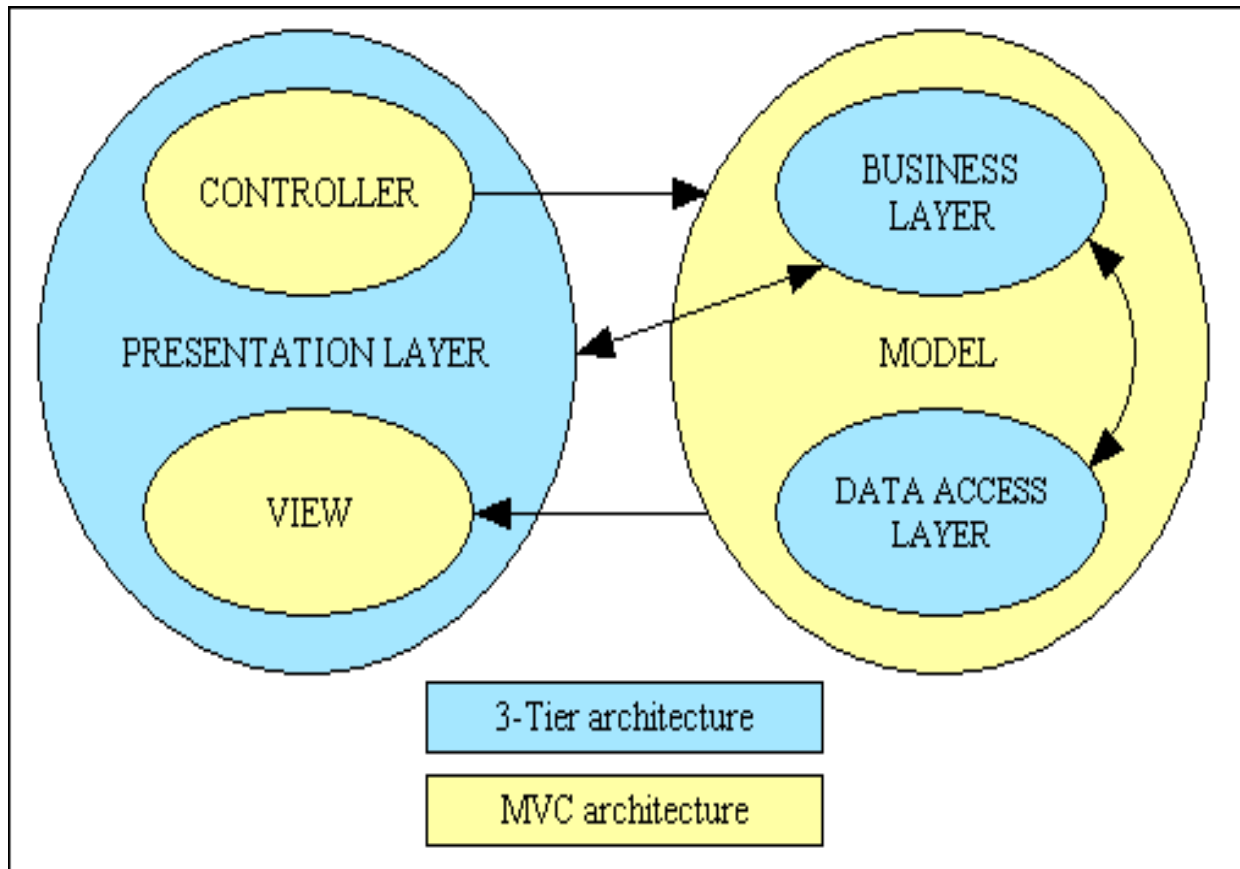
- High Performance
- High Scalability
- Low Complexity
- Easy to develop (from developer standpoint)
- Rich user experience (from user standpoint)
- Standards-based
- Vibrant ecosystem
- Multi-device support
- Mobile friendly

Web App Architecture Evolution

- Model 1 MVC (Template based)
- Model 2 MVC (Controller based)
 - > View is generated by the server per request
 - > Struts, SpringMVC, Rails
- Component oriented
 - > View is generated by the server per request
 - > JSF, Tapestry
- HTML5-enabled Rich Client/Thin Server
 - > View is generated at the client
 - > Client Model sync'ed with backend data via REST
 - > JavaScript MVC framework (Backbone.js, Angular.js)
 - > HTML 5 enabled

HTML5-enabled Rich Client Architecture

- Presentation (view generation) is done at the client



HTML5-enabled Rich Client Architecture

- All presentation handling is done on the client
 - > HTML5 semantics, JavaScript, CSS
- All conversational state (also known as view or presentation state) is kept on the client
 - > The server is stateless from the perspective of requests
 - > HTML5 offline technologies
- The client is responsible for fetching the data from the server and insert it into the presentation and extracting data from the presentation to send to the server
 - > HTML5 Web sockets, XHR2
 - > REST

Improved Performance

- Performance aspects includes
 - > server load
 - > bandwidth use
 - > latency
- The server load & bandwidth use is reduced - No time spent on server-side page generation and then transport it to the client
- The user response time is improved – the client is more responsive to user interactions

Scalability

- Moving the presentation to the client also moves all the conversational state to the client.
 - > This leverages the processing power of the clients.
 - > It distributes the per-client session memory and processing requirements inherent in server side architectures to all the clients.
- Transferring less data and not doing presentation generation should free up the server for handling more clients.
- It should be easier to add additional servers
 - > No (or less) server state needs to be replicated

Reduced Complexity

- The main reason complexity is reduced is because control of the UI is in one place rather than split between client and server
 - > The server is no longer micro managing the client by remote control.
 - > UI events stay on the client.
- Also the framework code that supports the application presentation layer can be implemented in one language (JavaScript) rather than for example JavaScript on the client and Java on the server.

Improved User Experience

- More responsive to user actions
- Offline usage of the application is possible
 - > It is possible since the UI handling is performed on the client

Is HTML5-enabled Rich Client Architecture Ready?

- Absolutely!
 - > HTML5 is now pervasive
 - > Highly popular JavaScript libraries – jQuery
 - > Emergence of client JavaScript MV* frameworks – Angular.js, Backbone.js
 - > JavaScript template – Mustache.js
 - > JavaScript engine is quite powerful now, thus can handle work load on client (and on the server)
 - > REST

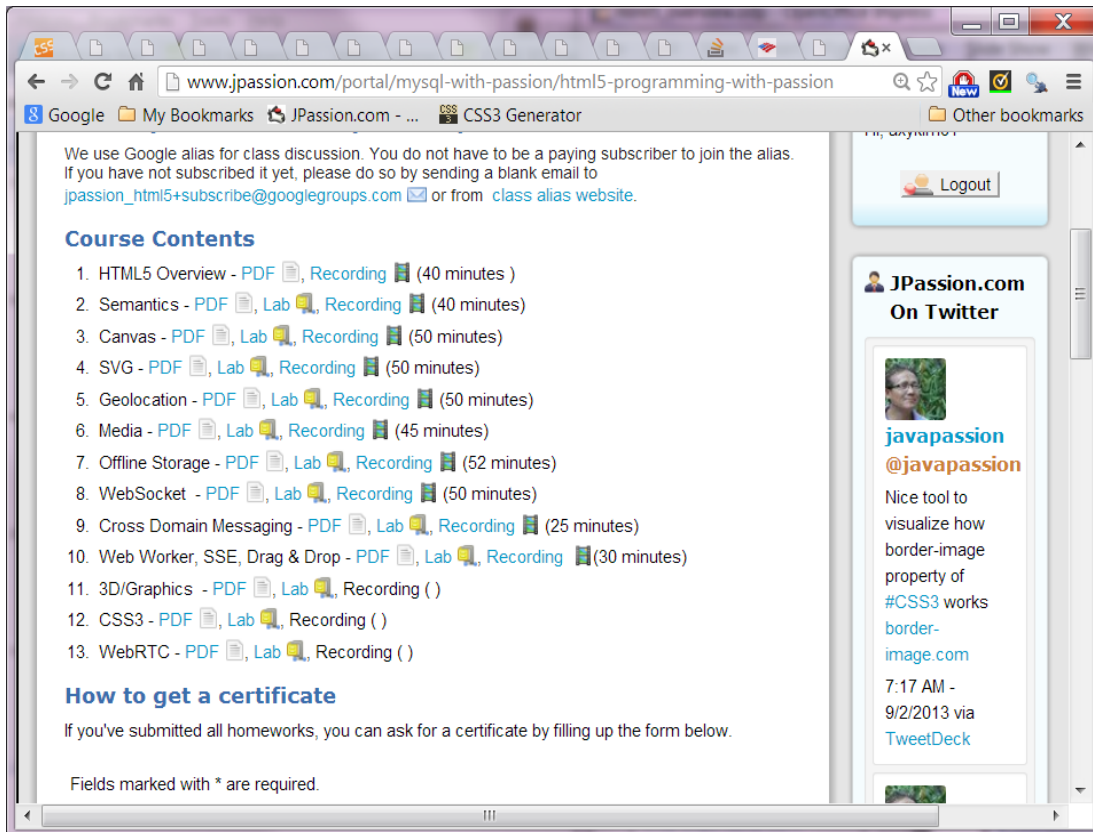
**So How Can I Get
Started?**

Things a Developer need to learn

- Frontend
 - > JavaScript, CoffeeScript
 - > HTML5 APIs (we learned today)
 - > jQuery
 - > Mustache.js
 - > JavaScript MVC framework (AngularJS)
- Backend
 - > Building RESTful services
 - > Use HTML5 friendly frameworks – NodeJS, Play, Grails, (Rails)
- Tool
 - > NetBeans 7.3+ now supports HTML5 app development

HTML5 Course from JPassion.com :-)

- <http://jpassion.com>
- We cover pretty much everything HTML5 in hands-on style



The screenshot shows a web browser window displaying the JPassion.com course page. The address bar shows the URL: www.jpassion.com/portal/mysql-with-passion/html5-programming-with-passion. The page content includes a notice about Google alias for class discussion, a 'Logout' button, and a 'Course Contents' section with 13 items. Each item lists the topic, available resources (PDF, Lab, Recording), and duration. Below the course contents is a 'How to get a certificate' section with a note about submitting homeworks and a requirement notice. On the right side, there is a 'JPassion.com On Twitter' widget showing a tweet from @javapassion about a CSS3 border-image property.

We use Google alias for class discussion. You do not have to be a paying subscriber to join the alias. If you have not subscribed it yet, please do so by sending a blank email to jpassion_html5+subscribe@googlegroups.com or from [class alias website](#).

Course Contents


1. HTML5 Overview - PDF, Recording (40 minutes)
2. Semantics - PDF, Lab, Recording (40 minutes)
3. Canvas - PDF, Lab, Recording (50 minutes)
4. SVG - PDF, Lab, Recording (50 minutes)
5. Geolocation - PDF, Lab, Recording (50 minutes)
6. Media - PDF, Lab, Recording (45 minutes)
7. Offline Storage - PDF, Lab, Recording (52 minutes)
8. WebSocket - PDF, Lab, Recording (50 minutes)
9. Cross Domain Messaging - PDF, Lab, Recording (25 minutes)
10. Web Worker, SSE, Drag & Drop - PDF, Lab, Recording (30 minutes)
11. 3D/Graphics - PDF, Lab, Recording ()
12. CSS3 - PDF, Lab, Recording ()
13. WebRTC - PDF, Lab, Recording ()

How to get a certificate

If you've submitted all homeworks, you can ask for a certificate by filling up the form below.

Fields marked with * are required.

JPassion.com On Twitter

 **javapassion**
@javapassion

Nice tool to visualize how border-image property of #CSS3 works border-image.com

7:17 AM - 9/2/2013 via TweetDeck

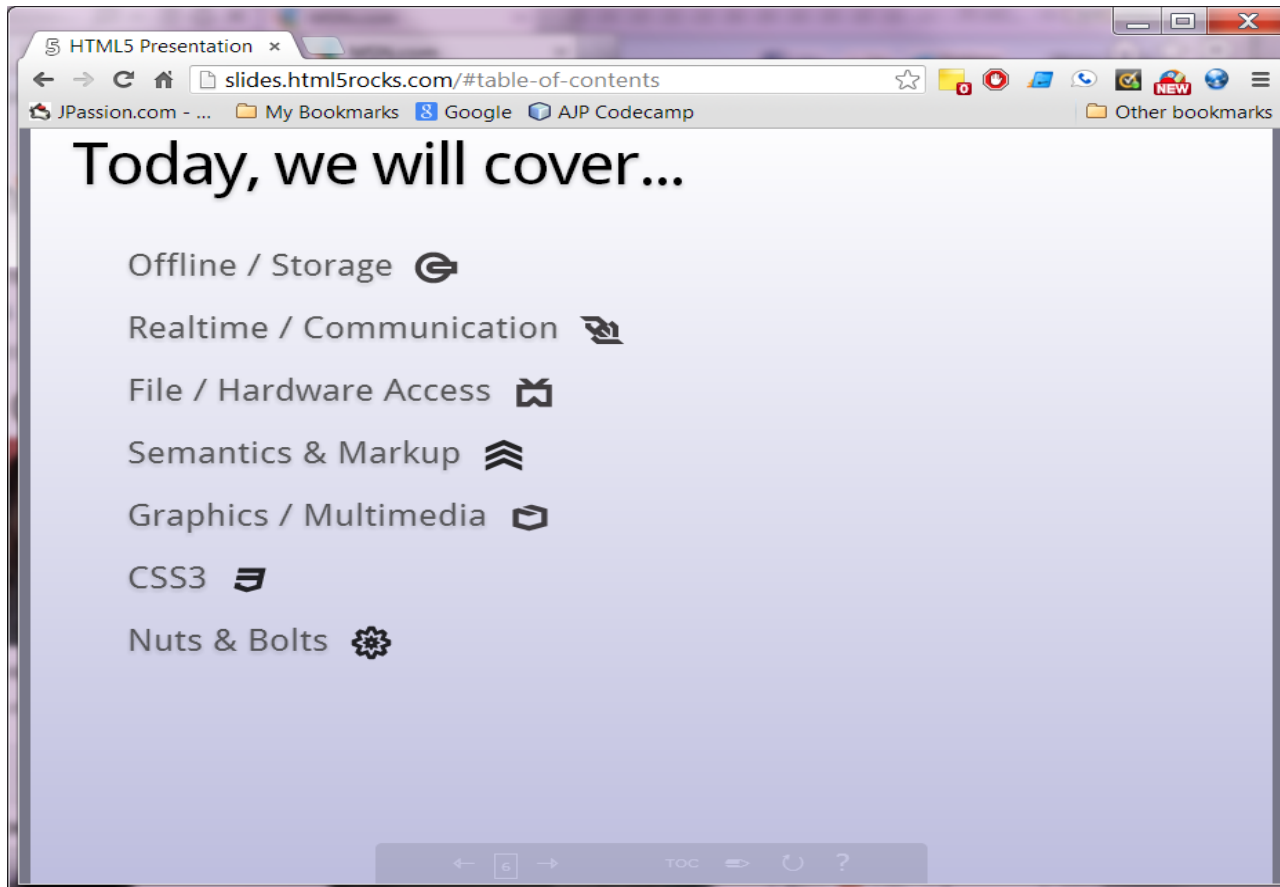
HTML Simple Tutorial Site #1

- <http://playground.html5rocks.com/>
- You can change and test code



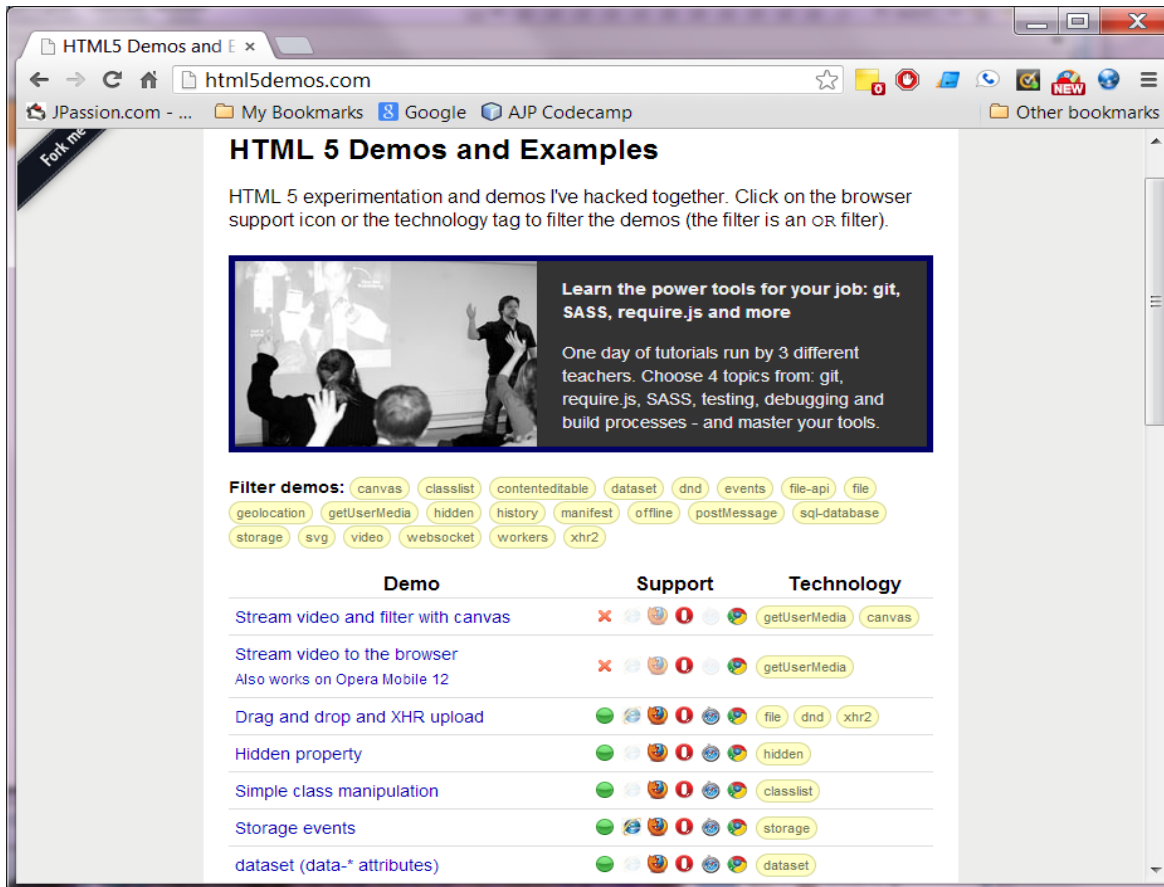
HTML Simple Tutorial Site #2

- <http://slides.html5rocks.com/#table-of-contents>



HTML Simple Tutorial Site #3

- <http://html5demos.com/>

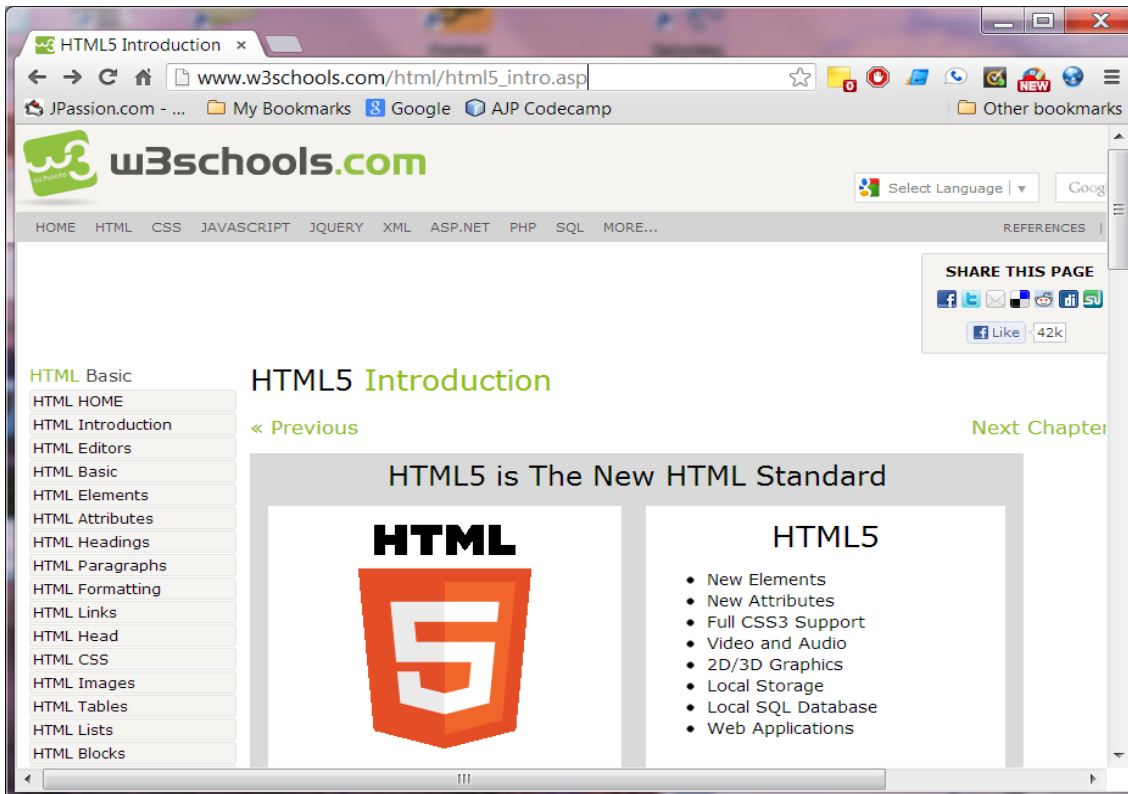


The screenshot shows a browser window with the URL html5demos.com. The page title is "HTML 5 Demos and Examples". Below the title, there is a paragraph: "HTML 5 experimentation and demos I've hacked together. Click on the browser support icon or the technology tag to filter the demos (the filter is an OR filter)." Below this is a featured article with a dark background and white text: "Learn the power tools for your job: git, SASS, require.js and more". The article text says: "One day of tutorials run by 3 different teachers. Choose 4 topics from: git, require.js, SASS, testing, debugging and build processes - and master your tools." Below the article is a "Filter demos:" section with various tags in yellow rounded rectangles: canvas, classlist, contenteditable, dataset, dnd, events, file-api, file, geolocation, getUserMedia, hidden, history, manifest, offline, postMessage, sql-database, storage, svg, video, websocket, workers, xhr2. Below the filters is a table with three columns: "Demo", "Support", and "Technology".

Demo	Support	Technology
Stream video and filter with canvas		getUserMedia canvas
Stream video to the browser Also works on Opera Mobile 12		getUserMedia
Drag and drop and XHR upload		file dnd xhr2
Hidden property		hidden
Simple class manipulation		classlist
Storage events		storage
dataset (data-* attributes)		dataset

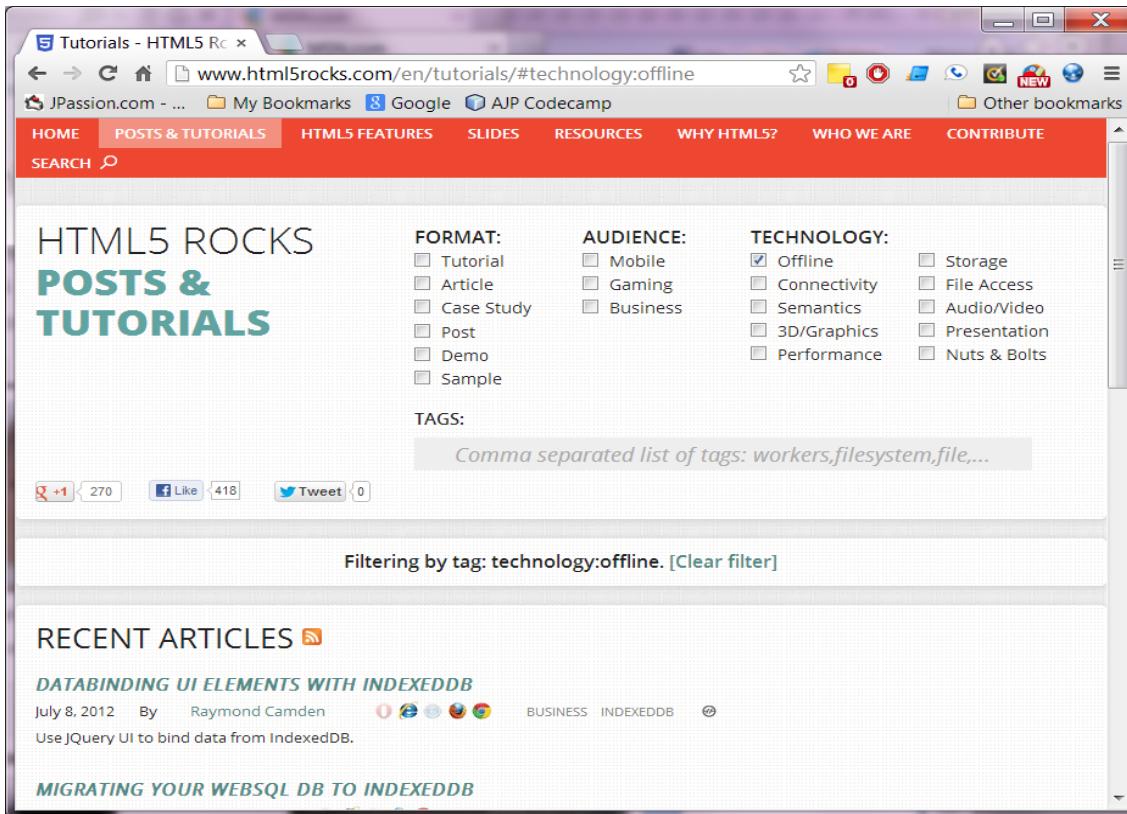
HTML Simple Tutorial Site #4

- http://www.w3schools.com/html/html5_intro.asp
- Focused mostly on HTML markup



HTML Advanced Tutorial Site

- <http://www.html5rocks.com/en/tutorials/>
- Community driven advanced tutorials



Learn with Passion!
JPassion.com

